Techniques d'analyse dynamique des média sociaux pour la relation client

Duc Kinh Le Tran

▶ **To cite this version:**

Duc Kinh Le Tran. Techniques d'analyse dynamique des média sociaux pour la relation client. Intelligence artificielle [cs.AI]. Télécom Bretagne; Université de Bretagne Occidentale, 2015. Français. NNT : . tel-01271823

HAL Id: tel-01271823
https://hal.science/tel-01271823

Submitted on 9 Feb 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE / Télécom Bretagne
sous le sceau de l'Université européenne de Bretagne

pour obtenir le grade de Docteur de Télécom Bretagne
En accréditation conjointe avec l'Ecole Doctorale Sicma
Mention : Informatique

présentée par
Duc Kinh Le Tran
préparée dans le département Logique des usages, sciences sociales et de l'information
Laboratoires : Orange Labs (Lannion) et Labsticc

Techniques d'analyse dynamique des média sociaux pour la relation client

Thèse soutenue le le 5 juin 2015
Devant le jury composé de :

Emmanuel Viennet
Professeur, Université Paris 13 / Président

Christine Largeron
Professeur, Université Jean Monnet IAE – Saint Etienne / Rapporteur

Patrick Gallinari
Professeur, LIP6, Université Pierre et Marie Curie / Rapporteur

Pascal Cheung-Mon-Chan
Ingénieur, Orange Labs - Lannion / examinateur

Cécile Bothorel
Maître de Conférences, Télécom Bretagne / Encadrant

Yvon Kermarrec
Professeur, Télécom Bretagne Directeur de thèse

N° d'ordre : 2015telb0353

Sous le sceau de l'Université européenne de Bretagne

Télécom Bretagne

En accréditation conjointe avec l'Ecole Doctorale Sicma

Techniques d'Analyse Dynamique de Média Sociaux pour la Relation Client

---

Thèse de Doctorat

Mention : **Informatique**

Présentée par **Duc Kinh Le Tran**

Département : **Logique des Usages, Sciences Sociales et de l'Information**

Laboratoire : **LabSTICC** Pôle : **CID**

Directeur de thèse **: Yvon Kermarrec**

Jury :

Mme Christine Largeron — Professeur, Université Jean Monnet (Rapporteur)
M. Patrick Gallinari — Professeur, LIP6, Université Pierre et Marie Curie (Rapporteur)
M. Emmanuel Viennet — Professeur, Université Paris 13 (Examinateur)
M. Yvon Kermarrec — Professeur, Telecom Bretagne (Directeur de thèse)
Mme Cécile Bothorel — Maître de Conférences, Telecom Bretagne (Encadrant)
M. Pascal Cheung-Mon-Chan — Ingénieur R&D, Orange Labs (Encadrant)

Techniques d'analyse dynamique de média sociaux pour la relation client

6 juillet 2015

REMERCIEMENTS

Je tiens d'abord à remercier mon directeur de thèse, Yvon Kermarrec (Directeur du Département Lussi, Télécom Bretagne), pour avoir accepté de diriger la thèse, pour sa disponibilité, pour les bons conseils concernant la méthodologie de recherche et pour la relecture du manuscrit de thèse.

J'aimerais exprimer ma profonde gratitude à Cécile Bothorel (Département LUSSI, Télécom Bretagne) pour ses travaux d'encadrement de cette thèse. Tout au long de la thèse, elle m'a toujours fait confiance, m'a guidé sur différentes thématiques de recherche. Grâce à elle, j'ai pu affiner le sujet de thèse et découvrir différentes pistes intéressantes. Elle m'a beaucoup aidé dans la rédaction des articles et le manuscrit de thèse. Merci beaucoup à Cécile pour sa constante disponibilité, ses encouragements, sa patience, sans lesquels je n'aurais jamais pu aller jusqu'au bout.

J'adresse également toute ma reconnaissance à Pascal Cheung-Mon-Chan, mon encadrant de thèse à Orange Labs, Lannion. Je remercie Pascal pour m'avoir proposé un sujet de thèse très intéressant et de m'avoir donné l'occasion de travailler dans l'équipe du data mining à Orange Labs. Merci à lui pour les conseils de méthodologie, pour les travaux de relecture des articles et du manuscrit de thèse et particulièrement ses énormes efforts qui m'ont permis de travailler dans les meilleures conditions.

Je tiens à remercier tous les membres de mon jury. Je remercie Patrick Gallinari (LIP6, Université Pierre et Marie Curie) et Christine Largeron (Université Jean Monnet) pour avoir accepté d'être rapporteurs de cette thèse. Un grand merci également à Emmanuel Viennet (Université Paris 13) pour avoir accepté de présider ce jury. Merci beaucoup pour l'intérêt que vous avez porté à cette thèse et pour les retours constructifs qui me permettrons d'améliorer mes travaux dans le futur.

Je tiens également à remercier tous les gens que j'ai pu rencontrer à Oranges Labs, Lannion. Je remercie particulièrement Fabrice Clérot et Tanguy Urvoy qui m'ont beaucoup aidé avec leur grande disponibilité et leur expertise technique. Merci à Nicolas Voisine et Bénédicte Cherbonnel qui m'ont aidé à obtenir et prétraiter les données d'Orange. Merci à Marc Boullé, Romain Trinquart pour leurs aides techniques concernant Khiops et leurs remarques intéressantes à propos de mes travaux. Merci à Anne Amsallem et Patrice Soyer d'avoir aidé à lancer cette thèse à Orange. Merci à Fabien Dupont d'avoir relu mon manuscrit et corrigé le français. Je remercie également les stagiaires, doctorants, post-doctorants à Orange Labs avec qui j'ai pu échanger et partager des moments agréables lors des pauses de café ou au

restaurant Sodexo. Merci à tous pour votre gentillesse, vos amitiés et vos aides pendant mon séjour à Orange Labs.

Mes remerciements vont également à tous les collègues du département Lussi, Telecom Bretagne. Merci pour leurs aides, leurs encouragements et une ambiance de travail très agréable dans le département. Je remercie particulièrement Romain Picot-Clemente, mon collègue de bureau à Telecom Bretagne, qui m'a partagé de bonnes expériences durant la thèse et m'a aidé à corriger le français dans le manuscrit.

Pour finir, je remercie de tout cœur ma famille et mes amis, et plus spécialement mes parents et ma copine, Uyen-Phuong. Je n'aurais jamais réussi sans leur soutien et leur confiance.

RÉSUMÉ

Cette thèse d'informatique en fouille de données et apprentissage automatique s'inscrit dans le contexte applicatif de la gestion de la relation client (*Customer Relationship Management* ou CRM). Avec l'émergence des média sociaux, une nouvelle philosophie de gestion de la relation client est apparue : le CRM Social. Le CRM Social se focalise sur l'engagement des clients et la prise en compte de plusieurs canaux de communication entre l'entreprise et les clients, y compris les média sociaux. Les entreprises perçoivent actuellement la nécessité d'une stratégie de relation client *intercanale* dans laquelle elles suivent le client à travers tout son parcours grâce à un système de canaux intégrés. L'objectif applicatif de la thèse est de concevoir de nouvelles méthodes de fouille de données adaptées à ce nouveau type de stratégie de relation client intercanale. Les techniques conçues dans la thèse permettent en effet de prédire les comportements du client à partir des données issues de multiples canaux. Nous nous intéressons aux comportements qui caractérisent l'engagement du client vis-à-vis de l'entreprise.

Nous effectuons d'abord une analyse des besoins dans laquelle nous montrons la nécessité des nouvelles techniques de fouilles de données pour une stratégie de relation client intercanale. Nous identifions les grands défis de la fouille de données, et parmi les défis les plus critiques que nous avons mis en évidence, nous nous intéressons au problème consistant à exploiter simultanément les informations de type attribut-valeur présentes dans les données clientèle d'une part et les informations liées aux interactions sociales (ou relationnelles) présentes dans les données riches et non structurées issues de média sociaux d'autre part.

Pour relever les défis posés dans la thèse, nous proposons de représenter les données issues de ces plusieurs canaux par un graphe complexe (un *réseau social attribué*) : un tel graphe est composé de différents types de nœuds. Les nœuds modélisent des clients-utilisateurs, mais aussi des mots qu'ils ont postés sur les média sociaux et les données attribut-valeurs clientèle ; le graphe contient plusieurs types d'arêtes également, représentant les interactions entre les nœuds de tout type. Nous introduisons une nouvelle méthode d'apprentissage incrémental basée sur les *modèles à facteurs latents*. Notre méthode, utilisant la factorisation de matrice, permet d'apprendre des facteurs latents sur les nœuds en exploitant notre réseau social attribué, intégrant ainsi simultanément le graphe social (représentant les interactions sociales) et les données attribut-valeurs clientèle de manière incrémentale. Les facteurs latents sont ensuite utilisés comme variables explicatives pour prédire les comportements des clients dans un processus d'apprentissage supervisé (dans notre cas nous utilisons la méthode de machines à vecteurs de support).

Nous effectuons ensuite des expérimentations sur des données synthétiques et réelles. Nous montrons que notre méthode de réduction de dimension est ca-

pable d'extraire des variables latentes informatives pour prédire les comportements des clients à partir des données intercanales. Sur le jeu de données synthétiques, nous montrons que notre méthode sait tirer partie de tous les types d'information. Elle s'avère particulièrement intéressante par rapport aux méthodes de référence dans le cas où la variable cible dépend fortement des données sociales. Elle capture efficacement le caractère d'homophilie selon lequel deux individus connectés dans un réseau s'influencent et ont tendance à adopter les mêmes comportements. Dans les expérimentations avec les données réelles, même si notre méthode produit des résultats du même ordre que les méthodes classiques, elle n'apporte cependant pas de gain en termes de performance (AUC) par rapport aux méthodes classiques basées sur la construction des variables explicatives caractérisant explicitement les interactions sociales. Dans le contexte applicatif particulier de la thèse, notre méthode ne présente pas d'intérêt opérationnel en l'état actuel. Cependant, son avantage principal est sa capacité à trouver les variables latentes informatives de manière automatique à partir de données complexes. Elle est donc utile pour d'autres champs d'application où il est difficile de chercher des variables explicatives informatives, comme par exemple dans un futur proche, les données volumineuses et hautement variables de l'Internet des Objets.

Dans les perspectives, nous proposons quelques pistes d'amélioration de notre méthode, notamment d'autres modèles à facteurs latents permettant d'exploiter différents types de corrélations entre les individus dans le graphe social.

Mots clés : *Customer Relationship Management, CRM intercanal, média sociaux, modèle à facteurs latents, factorisation de matrice*

Abstract

This thesis is in the field of data mining and in the context of *Customer Relationship Management (CRM)*. With the emergence of social media, CRM has become Social CRM, a new CRM philosophy which focuses on the *engagement* of customers through several channels of communication, including the social media. Companies today have seen the need for an *interchannel* (or *cross-channel*) strategy in which they keep track of their clients' histories through a consistent combination of multiple channels. The goal of this thesis is to develop new data mining methods to adapt to this new strategy of CRM. Consequently, the techniques developed in this thesis allow to predict customer behaviors using data collected from multiple channels. We are interested in all types of customer behaviors that characterized their *engagement* with respect to the company.

First of all, we perform a needs analysis in terms of data mining for interchannel CRM strategy. In this analysis, we point out to many new challenges for data mining and machine learning for this new strategy of CRM. Among these challenges, we focus on the problem of exploiting simultaneously both attribute-value information in the client data and relational (or social) information in data from social media. We are also interested in the incremental learning approach where we update the model with new data.

Next, we propose a new method of prediction of customer behaviors in the context of interchannel CRM. We propose to use a *social attributed network* to represent the data from multiple channels. This network contains multiple types of node : the social nodes represent social media users (or customers/clients), the attribute nodes represent both user-generated contents from social media (e.g words in their posts) and the attribute-value information from client data. The network also contains multiple types of edge representing the relations or interactions between all types of nodes. We introduce a new incremental learning method to learn latent factors on each node using matrix factorization. This method allows to learn latent factors, a low rank representation of the nodes, from social attributes network, in an incremental way. Thus, it is capable of exploiting simultaneously the social interactions, user-generated contents (from social media) and attribute-value information (client data). The latent factors are then used as features to predict customers' behaviors by training a supervised classifier (we use support vector machine in this case).

We then perform experiments on both synthetic and real data. We show that our method based on the latent factor models is capable of leveraging informative latent factors from interchannel data. On synthetic data, we show that our method is capable of leveraging useful information from social attribute networks. It is particularly interesting in case the target variable (customer behaviors) highly depends on the social graph. It effectively captures the *homophily* characteristic of the social graph, which states that two individuals connected in

the graph influence each other and tend to adopt the same behaviors. With real interchannel data, the proposed method has not shown any gain in prediction performance when comparing to the classic prediction methods which are based on manual feature construction on each individual from interchannel data. This means that our method is not yet relevant for the specific applicative context of this thesis. However, the main interest of our method is its capacity to find informative latent variables automatically from complex data, including social media. The method is useful for other applications where it is difficult to find informative features from data, for example when we have to deal with voluminous, highly dynamic data from the Internet of Things.

In future works, we consider some ways to improve the performance of our method, especially latent factor models that are able to leverage different types of relational correlation between individuals in the social graph.

Keywords : *Customer Relationship Management, cross-channel CRM, social media, latent factor models, matrix factorization*

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

LISTE D'ABRÉVIATIONS

CRM	Customer Relationship Mangagement
Social CRM	Social Customer Relationship Mangagement
FM	Factorisation de Matrice
RSA	Réseau Social Attribué
FRRM	Factorisation Régularisée Relationnelle de Matrices
FCM	Factorisation Collective de Matrices
MCA	Moindres Carrés en Alternance
AIMFL	Apprentissage Incrémental des Modèles à Facteurs Latents

CHAPITRE 1

INTRODUCTION

Sommaire

1.1 La gestion de la relation client à l'ère de média sociaux

1.1.1 La gestion de la relation client

Depuis longtemps, la gestion de la relation client (*Customer Relationship Management* ou *CRM*) est une partie essentielle dans les stratégies des entreprises. Classiquement, on définit le CRM comme l'ensemble de techniques et outils permettant de gérer les interactions entre une entreprise et ses clients. Les objectifs principaux d'un système CRM sont d'attirer de nouveaux clients, de conserver les clients que l'entreprise a déjà et de réduire le coût de marketing et de service à la clientèle [PS02]. Pour cela, une entreprise dispose de plusieurs canaux de communication et d'interaction avec ses clients (actuels ou potentiels) : points de vente, courrier, téléphone, e-mail, web, etc. En termes informatiques, les entreprises disposent d'un système, des outils et des techniques permettant d'une part d'interagir avec les clients et d'autre part de collecter, stocker, analyser les données clientèles. Parmi ces outils et techniques, les techniques de la fouille de données (*data mining*) sont parmi les plus importantes [RWY02].

1.1.2 Le rôle des média sociaux

L'avènement du Web Social dans les années récentes, qui met l'accent sur le partage de l'information et la collaboration des utilisateurs, a beaucoup influencé les stratégies de gestion de la relation client des entreprises. Les consommateurs deviennent plus actifs en participant aux sites web collaboratifs comme les forums, blogs, réseaux sociaux, plate-forme de partage multimédia, etc., collectivement désignés par l'appellation *média sociaux*. Dans ces nouveaux média, n'importe qui peut poster des commentaires et des avis sur les sociétés et leurs produits, qui peuvent influer les perceptions et les comportements d'achat d'un grand nombre d'autres utilisateurs du Web. Les conversations sur les valeurs d'une entreprise sont en mode *many-to-many* et dans la plupart de cas, il n'y a pas de participation de l'entreprise aux conversations. Les entreprises ne peuvent plus contrôler les conversations de leurs clients.

Avec l'émergence des média sociaux, les marques ont besoin de nouvelles stratégies et de nouveaux outils pour aborder la relation client. L'ensemble de ces stratégies et outils est souvent désigné par le terme *Social CRM (SRM)* en anglais (CRM social en français) pour distinguer avec le CRM traditionnel. Paul Greenberg, autorité reconnue dans le domaine de CRM, a donné une définition du Social CRM [Gre09] : « Le Social CRM est une philosophie et une stratégie d'entreprise, reposant sur une plate-forme technologique, des règles, des processus et des caractéristiques sociales. Son objectif est de se concentrer sur l'*engagement* avec les consommateurs au travers des conversations collaboratives, afin de créer des bé-

néfices pour les deux parties dans un environnement de confiance et de transparence. Le Social CRM est la réponse de l'entreprise à la prise de pouvoir du consommateur sur la conversation. »

D'après cette définition, le CRM social n'est pas simplement une intégration des média sociaux aux systèmes CRM traditionnels : il s'agit d'une évolution dans toute la philosophie du CRM. On peut citer quelques points clés sur le Social CRM :

Objectif Construire la relation avec les clients et les communautés de clients, se concentrer sur *l'engagement* avec les clients (au lieu de se concentrer sur la vente et le marketing comme dans le CRM traditionnel).

Canaux de communication Prendre en compte non seulement les transactions *business-to-customer* (i.e les canaux traditionnels comme points de vente, boutiques) mais également les interactions *customer-to-customer*, *customer-to-prospect* via les média sociaux.

Approche Écouter et participer aux réseaux conversationnels à travers des média sociaux. Les clients sont positionnés au centre - ils sont les maîtres des conversations.

Le Social CRM est vraiment une tendance de fond. Les entreprises sont en train d'effectuer des recherches pour exploiter la valeur des consommateurs dans les média sociaux. Selon l'étude réalisée par IBM en 2011 [IBM11], depuis 2010, près de 80% des entreprises sont présentes sur les médias sociaux et une majorité d'entre elles les utilise à des fins de relation client. Ces études confirment la volonté de s'engager dans une stratégie de relation client sur les média sociaux.

1.2 Enjeux de la gestion de la relation client à l'ère de média sociaux

Dans les paragraphes suivants nous décrivons les deux nouveaux enjeux du CRM à l'ère de média sociaux : (1) l'utilisation des différents canaux de communication et (2) l'accent sur l'*engagement* avec les clients. Ces enjeux font partie des attentes d'Orange pour son plan stratégique d'ici à 2020 [Cun15].

1.2.1 Vers une stratégie du CRM intercanale

Avec l'apparition de l'internet et particulièrement des média sociaux, les clients disposent de plus en plus de canaux pour interagir avec les entreprises. L'usage des terminaux mobiles connectés (smartphones, tablettes) se développe, les clients peuvent accéder aux canaux média sociaux, Web n'importe quand, n'importe où. D'après une étude en 2013 [SN13], les français ont utilisé en moyenne 2,9 canaux de contacts différents (2,7 en 2012). La

multiplication des canaux a modifié radicalement les parcours des clients. D'après [BJF11], un parcours client typique est ceci : le client s'informe en premier lieu sur Internet, puis se déplace en boutique, avant de revenir comparer les prix sur Internet, et lorsqu'il se décide à acheter, il appelle le call center pour plus d'informations.

On peut imaginer le processus de décision d'achat d'un client de la manière suivante : après avoir découvert un nouveau produit sur le Web, il y effectue une recherche, il se renseigne sur un comparateur de prix en ligne, il peut aussi poser une question sur un forum en ligne pour avoir plus d'informations, il consulte les stocks en ligne d'un magasin près de chez lui, il va au magasin et décide d'acheter le produit, finalement il revient sur les média sociaux en ligne (Twitter ou Facebook) pour déposer un avis, en cas de problème il peut soulever des questions auprès d'autres clients sur un forum spécialisé voire y déposer sa propre solution.

Avec plusieurs canaux disponibles pour interagir avec leurs clients, les entreprises pour la plupart ont commencé par adopter une stratégie *multicanale*. Cette stratégie de relation client implique une utilisation de plusieurs canaux sans forcément de lien entre eux. En effet, dans une telle stratégie, les canaux sont gérés de manière indépendante. La gestion de chaque canal vise à atteindre son propre objectif. L'inconvénient principal de cette approche de CRM est le manque de coordination entre les canaux. Dans l'exemple ci-dessus, la stratégie multicanale ne permet pas de suivre les clients à travers des différents canaux pour analyser son processus de décision d'achat.

Dans les années récentes, les entreprises ont perçu la nécessité d'un modèle *intercanal* (ou *cross-canal*). Vanheem et al. [VCL11] définissent la stratégie intercanale comme « la stratégie qui consiste à éliminer les ruptures, quelle que soit leur nature (physique, émotionnelle, économique, cognitive...) lors des changements de canaux par un client tout au long d'une même expérience avec une enseigne. »Avec une stratégie intercanale, l'entreprise suit le client à travers tout son parcours grâce à un système de canaux intégrés pour lui offrir des interactions cohérentes en fonction de son historique.

Le modèle intercanal est une évolution nécessaire du modèle multicanal. Il vise à créer des synergies entre les différents canaux. Avec ce modèle, l'entreprise définit un objectif commun et gère tous les canaux simultanément. Revenons sur le processus de décision d'achat illustré au-dessus. Imaginons que la décision du client peut être influencée par plusieurs facteurs : les produits achetés dans le passé, les avis qu'il a regardés sur les média sociaux, les recommandations des amis dans les réseaux sociaux etc. Ces facteurs viennent de plusieurs canaux (données clientèles, réseaux sociaux en ligne, les forums en ligne). Avec une stratégie du CRM intercanale, l'enjeu est de tout analyser pour caractériser la décision d'achat du client.

1.2.2 La notion d'engagement

La notion d'engagement est indispensable à explorer pour mettre en œuvre le Social CRM. Différentes définitions et modèles de l'engagement (de point de vue commercial) ont été proposées. Nous adoptons dans ce travail la notion d'engagement de Kumar et al. [KAD+10]. Ils considèrent l'engagement d'un client avec l'entreprise comme l'ensemble d'interactions actives entre le client et l'entreprise ou avec d'autres clients et prospects. Ces interactions peuvent apporter une certaine *valeur* pour l'entreprise, qui reflète le niveau d'engagement du client. La valeur du client concerne non seulement la somme des profits attendus sur la « durée de vie »du client (à travers des actes commerciaux du client) mais aussi d'autres aspects comme par exemple l'influence du client vers d'autres clients, sa capacité de recommandation et d'apport de nouveaux clients (mesurée par les interactions entre le client et les autres clients ou prospects), sa capacité de co-création avec l'entreprise à travers des réactions, feedbacks ou propositions.

Dans le contexte du CRM intercanal, l'engagement du client concerne les comportements des clients dans tous les canaux. Sur le plan commercial (les canaux classiques), les actes commerciaux comme des achats, des changements d'offre sont des comportements classiques du client qui reflètent son engagement. Dans les médias sociaux, les activités des clients (et prospects) sont très intéressantes de point de vue de l'engagement défini ci-dessus. Ces activités n'apportent pas de profit direct pour l'entreprise, mais ils apportent d'autres valeurs du client : sa capacité de co-création, l'influence du client et sa capacité d'apport de nouveaux clients. Par exemple, le fait qu'un client donne un avis à propos d'un produit ou d'un service de l'entreprise sur un forum en ligne est compté dans l'engagement de ce client avec l'entreprise. Cet acte reflète à la fois la co-création (permet à l'entreprise de mieux comprendre leur client) et peut avoir un impact sur les autres clients et prospects via l'effet de bouche-à-oreille dans les média sociaux. D'une manière similaire, le fait qu'un client (ou client potentiel) poste un *tweet* à propos de l'entreprise est aussi un indicateur de l'engagement du client. Nous reverrons ce type d'activités du client dans le chapitre 4.

1.3 Les enjeux techniques d'une stratégie de relation client intercanale

Dans une stratégie multicanale, chaque canal dispose de ses propres outils et de ses propres bases de données. L'enjeu technique majeur du passage du multicanal vers l'intercanal est de centraliser et d'exploiter les données issues de différents canaux. Il s'agit de mettre toutes les données de plusieurs sources (ventes e-commerce, vente en boutiques, web, forum d'entraide, service après vente, etc.) dans des bases de données centralisées. Cela assure que

les clients, qui utilisent différents canaux, peuvent avoir une vision cohérente des produits et des services de l'entreprise et vice versa, l'entreprise peut avoir une vision globale de ses clients à travers de multiples canaux.

Concernant la fouille de données, avec le CRM intercanal, nous avons besoin de techniques avancées pour traiter conjointement les données de différentes natures, issues des différentes sources dans les bases de données intercanales. Dans le CRM traditionnel, les données sont souvent organisées en forme de tables (données *tabulaires*), où chaque enregistrement décrit les caractéristiques d'un client. Ces données sont prêtes pour appliquer les techniques de fouille de données classiques. Dans le CRM intercanal, les données issues des média sociaux, variées, potentiellement non-tabulaires, volumineuses, plus ou moins dynamiques, peuvent poser de nombreux défis pour la fouille de données. Le défi majeur pour notre problématique d'intercanalité est comment exploiter les *données sociales* (les relations, les interactions) issues des média sociaux conjointement avec les données tabulaires. Les données sociales sont en effet très riches : les liens d'amitié dans les réseaux sociaux en ligne, les échanges (messages), les participations dans les fils de discussion dans un forum, etc. Elles portent des informations utiles à exploiter pour le CRM. Par exemple, les clients connectés dans les média sociaux (e.g par des liens d'amitié) sont plus susceptibles d'effectuer les mêmes actions commerciales. La décision d'un client peut être influencée par les activités d'autres clients/internautes sur les média sociaux. Ainsi, la prise en compte des relations/interactions permet de capturer, par exemple, des phénomènes d'influence sur les média sociaux.

Une difficulté supplémentaire pour la prise en compte des interactions des clients sur certains média sociaux utilisés dans le cadre d'une stratégie de relation client intercanale comme un Forum d'entraide, est que les relations entre les utilisateurs de ces media sociaux peuvent avoir des caractéristiques différentes des réseaux sociaux basés sur des liens d'amitié ou affinitaires (comme Facebook) : par exemple sur un forum d'entraide on peut imaginer qu'un utilisateur novice sera aidé par un utilisateur expérimenté (ou même un employé de la marque pendant son temps libre), alors que ces deux catégories d'utilisateurs ont des profils différents. Autrement dit, on peut se demander si certaines hypothèses que l'on fait habituellement pour analyser des réseaux sociaux affinitaires comme Facebook sont bien adaptées pour l'analyse de média sociaux comme un Forum d'entraide.

Un autre défi est la prise en compte des *contenus* créés par les clients dans les média sociaux. Dans les média sociaux, en plus des données de type relations sociales, nous trouvons aussi les contenus créés par les utilisateurs : les profils, les textes (messages, commentaires, posts), les notes apposées à des produits, les centres d'intérêts déclarés par les utilisateurs, etc. Ces données sont souvent de grande dimension et creuses. Par exemple, les textes sont souvent utilisés dans la fouille de données en forme de matrice creuse à grande dimen-

sion (la matrice document-terme). L'utilisation efficace de ces données conjointement avec d'autres types de données intercanales est non triviale, et à notre avis, importante à considérer.

Enfin, nous mentionnons la prise en compte de la dynamique, dans le sens où les données sont datées et évoluent au fil du temps. Dans les média sociaux, de nouveaux types de contenus peuvent apparaitre au fil du temps. Par exemple, si on considère un mot (dans les textes des utilisateurs) comme un contenu, on peut imaginer que l'ensemble de mots utilisés dans les textes des utilisateurs peut évoluer. Dans les données clientèle de l'entreprise, les nouveaux attributs peuvent apparaître (par exemple, les nouveaux type de contrat, nouveaux actes commerciaux). De plus, les dépendances statistiques entre différents éléments des données intercanales, par exemple la dépendance entre les actes commerciaux des clients et les contenus publiés sur les média sociaux, peuvent changer au fil du temps. Les évolutions des données dans différents canaux ne sont pas dans la même temporalité. Par exemple, on peut imaginer que les données issues des média sociaux reflètent souvent les événements éphémères, et donc les données issues de média sociaux évoluent plus rapide que les données clients (qui concernent par exemple le changement d'offre, le type de services, la consommation, etc). Cela est aussi un aspect qu'il serait nécessaire de prendre en compte.

1.4 Objectif et problématiques

Cette thèse, qui s'inscrit dans le domaine de la fouille de données, vise à étudier et à concevoir de nouvelles méthodes d'analyse des données adaptées à une stratégie de relation clients intercanale. Nous nous sommes particulièrement intéressés à la prédiction de comportements des clients.

Ainsi, l'objectif principal de cette thèse est de concevoir des nouvelles techniques de fouille de données permettant de *prédire les comportements des clients* à partir des données intercanales.

La prédiction de comportements des clients est très importante dans la gestion de la relation client. Elle permet à l'entreprise d'avoir des réactions pro-actives vis-à-vis des clients à l'échelle individuelle. Par exemple, la capacité de prédire les actes commerciaux (achats, *churn*) permet aux entreprises de cibler les clients potentiels ou retenir les clients existants. A l'ère de média sociaux, avec la notion de l'engagement étendue (définie dans la section 1.2.2), nous chercherons à capter des *comportements intercanaux*. Les comportements intercanaux comprennent non seulement des actes commerciaux mais aussi les activités des clients sur les autres canaux (i.e média sociaux) qui reflètent l'engagement avec la marque. Le fait que le client parle de la marque, les avis positifs ou négatifs du client à propos d'un produit ou service... font partir des comportements que nous voulons prédire.

Dans cette thèse, nous allons d'abord identifier les comportements des clients qui caractérisent l'engagement ; il s'agit des actes commerciaux des clients ou les activités des clients dans les média sociaux (e.g le fait qu'un client parle de la marque, des produits de la marque). Nous chercherons à prédire ces comportements. Pour accomplir cet objectif, nous avons identifié des problématiques dans la thèse.

- exploiter conjointement les données tabulaires (issues de la base de données clientèles) et les données de type interactions sociales (issues des média sociaux). Cette problématique est une conséquence de la multiplication des canaux (comprenant les média sociaux) du point de vue de fouille de données. C'est le défi majeur de la fouille de données intercanales (comprenant les média sociaux) et a été identifié et décrit dans la section 1.3.
- intégrer les données de type « contenus »à grande dimension (e.g les textes) issues des média sociaux dans la modélisation. Les média sociaux contiennent une quantité importante de données variées pouvant rentrer dans cette catégorie, notamment les mots émis lors d'interaction. Le nombre de mots et d'individus différents est en effet potentiellement très grand dans un jeu de données d'interactions sociales.
- prendre en compte la dynamique des données, particulièrement les données issues des média sociaux. Par le terme « dynamique », nous voulons désigner l'évolution des caractéristiques des données au fil du temps. Un problème est de considérer l'apparition des nouveaux types de contenu créés par les utilisateurs dans les média sociaux au fils du temps. Les dépendances statistiques entre différents éléments des données intercanales peuvent aussi évoluer. A noter que les évolutions des données dans les différents canaux ne sont pas dans la même temporalité. La prise en compte de la dynamique des données, et ses différentes vitesses de changement, aide probablement à améliorer la performance de prédiction.
- la méthode conçue doit être efficace en termes de passage à l'échelle. C'est une exigence commune des techniques de fouille de données en CRM, étant donné que les données intercanales à traiter sont de plus en plus volumineuses. Les approches moins coûteuses en termes de calcul sont toujours plus intéressantes dans un contexte opérationnel.

1.5 Contributions et organisation du document

La première contribution de cette thèse est l'analyse des besoins dans laquelle nous montrons la nécessité des nouvelles techniques de fouilles de données pour une stratégie de relation client intercanale. Nous identifions aussi les grandes problématiques à résoudre (les sections précédentes) pour construire les stratégies de relation client intercanale de demain.

À partir de cette analyse, nous pouvons effectuer un état de l'art sur les travaux académiques connexes.

La contribution principale de cette thèse est la conception d'une nouvelle méthode de fouille de données adaptée à une stratégie de relation client intercanale pour prédire des comportements des clients. Cette méthode, nommée *Apprentissage Incrémental des modèles à facteurs latents* (AIMFL), est basée sur les modèles à facteurs latents. Elle est capable d'apprentissage incrémental, dans le sens où elle met à jour le modèle avec les données récentes. Nous proposons d'utiliser un graphe social enrichi avec les attributs pour représenter efficacement les données intercanales comprenant les média sociaux. Nous avons appliqué la méthode sur différents jeux de données (synthétiques et réels) pour tester la validité ainsi que le comportement de la technique proposée dans différentes situations.

La suite du document est organisée est organisée en 4 chapitres.

- Le **Chapitre 2** fait l'état de l'art sur les techniques de fouille de données pour la gestion de la relation client à l'ère de média sociaux. La **première section** de ce chapitre présente les techniques de fouille de données client (i.e les données stockées dans le système d'information de l'entreprise). Ce sont les techniques classiques (e.g la classification, le clustering) qui exploitent les données tabulaires. La **deuxième section** présente les techniques de fouille de média sociaux pour la relation client. Les techniques de fouille de média sociaux sont très variées, ici nous présentons quelques techniques typiques et ses applications en CRM : les mesures de l'influence, le marketing viral et le monitoring de média sociaux. La **troisième section** présente les techniques de l'*apprentissage statistique relationnel*, qui sont intéressantes pour la fouille de données intercanales parce qu'elles sont capables d'exploiter simultanément les données sociales (relations) et les données tabulaires.
- Le **Chapitre 3** présente une nouvelle méthode de prédiction des comportements avec les données intercanales - la contribution principale de la thèse. Dans les **deux premières sections** de ce chapitre, nous révisons brièvement les approches d'apprentissage automatique liées à notre méthode : les modèles à facteurs latents et l'apprentissage incrémental. Dans la **troisième section**, nous introduisons la notion de *réseau social attribué (RSA)*, la représentation de données intercanales que nous utilisons dans notre algorithme d'apprentissage. La **quatrième section** reformule le problème de la thèse avec la nouvelle représentation de données. L'algorithme d'apprentissage de notre méthode est décrit dans la **cinquième section**. La **sixième section** présente les premières expérimentations avec la méthode proposée sur les jeux de données synthétiques.
- Le **Chapitre 4** présente les expérimentations correspondant à deux cas d'usage de la méthode proposée :

- La **première section** présente les expérimentations sur des données que nous avons recueillies à partir de Twitter. L'objectif est de prédire qui parlera de la marque (Sosh), en exploitant simultanément les liens sociaux entre les individus et les textes (dans les *tweets*) sur Twitter.
- Dans la **deuxième section**, nous présentons les expérimentations sur un jeu de données intercanales anonymisées. L'objectif est de prédire des actes commerciaux des clients.

– Enfin, le **Chapitre 5** fait un bilan sur les travaux menés dans la thèse et les apports de la thèse. Ce chapitre présente aussi les perspectives de travaux futurs.

En somme, dans cette thèse, à partir des enjeux de la fouille de données dans le contexte de Social CRM, nous avons effectué les étapes suivantes : analyse des besoins, identification des problématiques, état de l'art, proposition des nouvelles techniques et évaluation expérimentale. Les travaux menés dans cette thèse établissent les premiers pas vers l'analyse de données dans le cadre d'une stratégie de relation client intercanale - nouvelle stratégie de relation client à l'ère de média sociaux. Les techniques conçues permettent de capter l'engagement des clients avec l'entreprise - un point clé du Social CRM.

CHAPITRE 2

LES TECHNIQUES EXISTANTES APPLICABLES À UNE STRATÉGIE DE RELATION CLIENTS INTERCANALE

Sommaire

Comme mentionné dans le Chapitre 1, les techniques de la fouille de données sont très importantes pour le CRM, en particulier le Social CRM à l'ère de média sociaux. Dans ce chapitre, nous identifions les techniques existantes de la fouille de données (y compris les données issues des média sociaux) pour la gestion de la relation client. L'idée est d'examiner les techniques connues de la fouille de données et de la fouille de média sociaux et d'évaluer leur potentiel dans le contexte de la thèse. Cette étude bibliographique permet de positionner nos problématiques dans un cadre académique.

2.1 La fouille de données client et le CRM

La fouille de données (*data mining*) a été utilisée depuis longtemps pour le CRM [BST99, RWY02]. Les entreprises collectent et stockent les données sur leurs clients actuels ou clients potentiels. Ces données contiennent par exemple les tables qui décrivent les clients (e.g les profils clientèles, données de facturation), les sondages sur un sous-ensemble des clients et prospects (qui répondent à des questions détaillées), les données comportementales des clients (e.g mode de paiement, interactions avec le service après vente). Avec les outils et techniques de la fouille de données, les entreprises peuvent découvrir les connaissances cachées dans ces données. Par exemple, depuis 2009, une plate-forme industrielle de ciblage de la clientèle développée à Orange Labs a été capable de construire des modèles prédictifs pour les jeux de données ayant un très grand nombre de variables d'entrée (des dizaines de milliers) et des instances (des dizaines de milliers) [FBC+10].

Dans [NXC09], on peut trouver une liste d'applications classiques de la fouille de données en CRM. Par exemple, les entreprises peuvent identifier les clients les plus importants, prédire les comportements futurs des clients, chercher les groupes de clients ayant des caractéristiques communes, etc. La fouille de données permet aux entreprises de prendre des décisions pro-actives dans leur processus du CRM. Les techniques de fouilles de données souvent utilisées sont la classification, le clustering, la fouille de séquence, les règles d'association et la visualisation. Ici nous citons quelques applications importantes des grandes techniques de la fouille de données pour le CRM.

2.1.1 Les techniques de clustering

Les techniques de *clustering* [Ber05] consistent à regrouper un ensemble d'objets de manière que les objets dans le même groupe (appelé *cluster*) sont plus proches (dans un certain sens) que ceux des groupes différents. Dans la perspective de l'apprentissage automatique, les clusters (qui ne sont pas prédéfinis) correspondent aux caractères cachés des données, et la recherche des clusters est un apprentissage *non-supervisé*.

En CRM, le clustering est souvent utilisé pour la segmentation de la clientèle (*customer segmentation*). Cette dernière joue un rôle important dans l'identification des clients en regroupant les clients similaires. Lefait et al.[LK10] ont montré qu'en utilisant les techniques de clustering sur les données d'achat des clients, on peut avoir un aperçu utile de la clientèle de l'entreprise et découvrir des tendances intéressantes des clients. Sankar [Raj11] utilise le clustering pour identifier les groupes des clients générant des profits élevés, des grandes valeurs et à faible risque. A partir des données clientèles, il a réussi à trouver un cluster représentant généralement 10-20 % des clients et qui donne 80% du chiffre d'affaires. Citons d'autres exemples d'application du clustering pour la segmentation de la clientèle : [CNP03, CHH07, ZGH09, HK12]. Dans tous les cas, le clustering aide les entreprises à avoir une meilleure vision sur leur clientèle.

Les techniques de clustering sont parmi les outils les plus utilisés dans CRM. Ces techniques ne répondent pas directement à l'objectif de la thèse. Elles ont pour but de détecter les groupes homogènes de clients (les clients ayant les comportements similaires), alors que dans cette thèse nous voulons prédire des comportements futurs des clients.

2.1.2 Les techniques de classification

Comme le clustering, la classification est l'un des modèles d'apprentissage les plus importants de la fouille de données. La classification consiste à prédire l'appartenance à un groupe (classe, label) pour un ensemble d'objets. Différente du clustering, la classification est un apprentissage *supervisé*. Son principe est d'apprendre un modèle à partir de données de l'apprentissage dans lesquelles les labels des objets sont connus et appliquer le modèle sur les nouveaux objets (données de test) pour prédire les labels de ces nouveaux objets.

En CRM, les techniques de classification aident à construire des modèles prédictifs pour différents types de comportement des clients, par exemple abandonner le service de l'entreprise (*churn*), acheter un produit ou s'abonner à un service (appétence), mettre à niveau un abonnement ou souscrire à une option (*up-selling*). Le problème le plus connu est la prédiction des *churns* (les pertes de clients), particulièrement dans le secteur de télécommunication. La prédiction des churns aide à identifier les clients les plus susceptibles de *churner* et donc permet à l'entreprise d'effectuer les actions de marketing appropriées pour retenir ces clients. Beaucoup de travaux [WC02, AMK04, XJ08, MQ09, CS09, DBG$^+$09, LC06, IRK12] ont utilisé la classification supervisée pour la prédiction des churns avec les données clients (transactions, facturations, etc.). L'idée est d'apprendre un modèle de classification (churn - non churn) à partir des données du passé et de le déployer sur les données actuelles pour prédire les futurs churns. Ce processus est illustré dans la figure 2.1. Les techniques de classification les plus utilisées sont : arbre de décision, forêt aléatoire, machines à vecteurs de support (*support vector machine* ou *SVM*) etc.

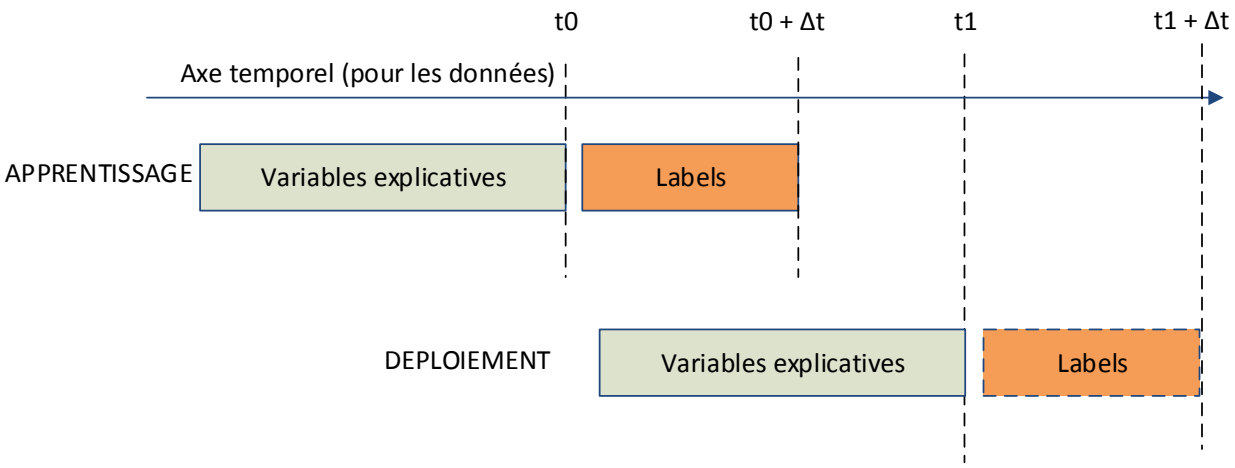


FIGURE 2.1 – La prédiction des variables cibles (e.g *churns*) avec la classification supervisée.

2.1.3 Autres techniques de la fouille de données pour le CRM

La régression est aussi un modèle d'apprentissage supervisé. Différente de la classification, la régression consiste à prédire une valeur réelle sur les objets. Verhoef et Donkers [VD01] ont utilisé la régression linaire pour prédire des *valeurs potentielles* des clients. La valeur potentielle d'un client est définie comme le bénéfice total sur tous les achats de ce client. La possibilité de prédire la valeur potentielle permet à l'entreprise d'effectuer des stratégies spécifiques sur ses clients. De la même façon, la régression (avec les forêts aléatoires) a été utilisée pour prédire les valeurs des clients dans [LV05, LCCL06].

A part le clustering, la classification et la régression, on peut citer d'autres techniques de fouille de données pour le CRM. Par exemple, la recherche de règles d'association, qui consiste à découvrir des relations intéressantes entre les variables dans une grande base de données, est souvent utilisée dans le *market basket analysis* pour découvrir des produits ou services que les clients achètent ensemble [APY02, BSVW04, JN06]. La fouille de séquence (*sequence discovery*, la découverte des motifs pertinents dans les données séquentielles) a été aussi utilisée pour analyser les comportements (e.g comportements d'achat, churns) des clients [CWLL03, VB05, CM06].

Parmi les techniques de fouille de données que nous avons citées dans cette section, les techniques d'apprentissage supervisé sont les plus pertinentes pour la thèse. Ces techniques permettent de prédire des actes commerciaux futurs (avec la classification) ou les valeurs potentielles des clients (avec la régression). Pourtant, ces techniques sont applicables sur les données attribut-valeur. Les données sociales issues des média sociaux doivent être transfor-

mées en données tabulaires avant d'appliquer ces techniques. La transformation de données sociales en données tabulaires consiste à construire un ensemble de variables explicatives à partir des données sociales. Ce processus exige d'avoir une expertise pour identifier quelles sont les variables informatives à construire pour prédire la variable cible (processus itératif test-erreur).

2.2 La fouille de données issues des média sociaux pour le CRM (ou social CRM)

Il y a récemment une quantité remarquable de recherches sur les techniques d'analyse de média sociaux. Nous présentons dans cette section les techniques et les approches d'analyse de média sociaux souvent trouvées dans le processus du Social CRM. A part les modèles de la fouille de données classiques, la majorité des autres techniques sont basées sur un graphe (ou réseau social) où les nœuds (sommets) sont des individus et les liens sont des relations sociales entre ces individus. Un graphe est une représentation naturelle des données issues des média sociaux. Une synthèse des applications actuelles ou potentielles de la fouille de réseaux sociaux en CRM (particulièrement en marketing) se trouve dans [MG10, BCGJ11].

2.2.1 Mesure de l'influence

De nombreux travaux modélisent et mesurent l'influence ou l'importance des nœuds dans un réseau. La possibilité de mesurer l'influence est très utile pour le CRM : elle permet d'identifier les clients « importants »dans un réseau de clients, ou les personnes à cibler dans une campagne de marketing.

On peut citer d'abord le célèbre Klout [1], une application en ligne permettant de mesurer l'influence d'un utilisateur à travers leur réseau social. L'analyse est effectuée sur les données provenant de sites tels que Twitter [2] et Facebook [3] et mesure la « taille »du réseau d'une personne, le contenu créé, et comment d'autres personnes interagissent avec ce contenu. Malgré la controverse autour de l'exactitude de la mesure de Klout, il a été utilisé par certains professionnels de médias sociaux comme un baromètre de l'influence.

La question de mesurer l'influence dans un réseau social a été étudiée depuis très longtemps dans l'analyse de réseaux sociaux [Fre79, WF94]. Dans la théorie de l'analyse de réseaux sociaux, nous pouvons trouver les mesures sur les nœuds qui s'appellent les « centralités »[MG10]. Les mesures de centralité indiquent l'importance d'un acteur au sein du réseau

1. http ://klout.com/home
2. https ://www.twitter.com/
3. https ://www.facebook.com/

social ; les exemples très simples sont les suivants : centralité de degré (nombre de relations dans lesquelles un acteur est engagé), centralité de proximité (nombre d'individus par lequel l'acteur doit passer pour entrer en contact avec les autres), centralité d'intermédiarité (nombre de fois où un nœud se trouve sur le chemin d'une paire de nœuds).

Le célèbre PageRank [PBMW99] peut être considéré comme une mesure de centralité. PageRank est un algorithme itératif d'analyse de liens permettant de calculer le rang de chaque nœud dans un graphe directionnel. Il est initialement conçu pour mesure quantitativement la popularité d'une page web. PageRank est basé sur l'hypothèse suivante : le rang d'une page web est d'autant plus élevé qu'elle est pointée par des pages avec des rangs élevés. Dans le même contexte (de classement sur des pages web), Kleinberg [Kle99] a introduit HITS (*Hyperlink-Induced Topic Search*), un algorithme itératif pour calculer à la fois deux mesures : le *hub score* (*l'hubitude*) et l'*authority score* (*l'autoritude*). Intuitivement, un nœud a une autoritude élevée s'il est pointé par un nœud ayant une hubitudes élevée ; un nœud a une hubitude élevée s'il est pointé par un nœud ayant une autoritude élevée. Dans la théorie de l'analyse de réseaux sociaux, le rang (*PageRank*), l'autoritude et l'hubitude (*HITS*) sont considérés comme les variantes de la centralité de vecteur propre (*eigenvector centrality*). On peut calculer ces mesures par le calcul de valeurs et de vecteurs propres d'une matrice.

Les mesures de l'influence sont intéressantes pour le Social CRM. On peut considérer ces mesures comme des métriques caractérisant les clients. Ces mesures peuvent être utilisées directement, par exemple pour identifier les clients/prospects « clés » dans les média sociaux. Weng et al. [WLJH10] a adapté le PageRank pour identifier les personnes les plus influentes (en fonction d'un sujet donné) dans le réseau social Twitter. De la même façon, Lam et Wu [LW09] ont adapté PageRank et développé *BuyerRank* pour trouver les acheteurs potentiels sur e-Bay. HITS a été utilisé dans le travail de Jurczyk et Agichtein [JA07] pour trouver les demandeurs/répondeurs pertinents dans un forum d'entraide.

Ces mesures ont été aussi utilisées pour prédire des comportements des clients, comme dans [BV12]. Dans ce travail, les mesures de l'influence (les centralités issues de l'analyse de réseau social) dans le réseau de parenté des clients (*customer kinship network*) sont utilisées comme des variables explicatives pour prédire le churn. L'idée proposée dans [BV12] est intéressante pour la thèse : les mesures de l'influence peuvent être utilisées comme variables explicatives conjointement avec les autres variables (issues des autres canaux) pour prédire des actes commerciaux des clients. Cette idée retombe dans l'approche nous avons mentionnée dans la section précédente : utiliser la classification supervisée pour prédire les comportements des clients. Ici, l'utilisation de mesures de centralité est une manière de transformer les données sociales en données attribut-valeur.

2.2.2 Modèle de l'influence et marketing viral

Il existe beaucoup de travaux qui étudient les modèles de l'influence dans un réseau social. Le but est de modéliser le processus de la diffusion de l'information, la propagation d'une nouvelle idée ou l'adoption d'un service ou produit dans un réseau social, avec l'hypothèse qu'un nœud est influencé par les nœuds de son voisinage. Dans les modèles de l'influence, on considère que chaque nœud est dans un statut soit *actif* soit *inactif*. Une fois qu'un nœud est actif, il peut activer ses nœuds voisins. Les deux modèles le plus connus sont *Linear Threshold Model* et *Independent Cascade Model* [Wor08, EK10].

Dans l'*Independent Cascade Model*, quand un nœud v devient actif à l'instant t, il est considéré comme contagieux. Il a une chance d'influencer chaque voisin u inactif avec une probabilité $p_{v,u}$, indépendamment de l'histoire jusqu'à présent. Si la tentative réussit, u devient actif à l'instant $t+1$. La probabilité $p_{v,u}$ peut être considérée comme la force de l'influence de v sur u.

Dans le *Linear Threshold Model*, chaque noeud u est influencé par chacun de ses voisins v selon un poids $p_{v,u}$ de telle sorte que la somme des poids entrants vers u ne dépasse pas 1. Chaque nœud u choisit un seuil uniforme θ_u au hasard dans l'intervalle $(0, 1)$. Si à l'instant t la somme des poids des voisins actifs d'un nœud u inactive est supérieur à θ_u alors u devient actif à l'instant $t+1$.

L'application principale des modèles de l'influence en CRM est le marketing viral. L'idée derrière le marketing viral est la suivante : en se basant sur un modèle d'influence, on peut cibler les utilisateurs les plus influents dans le réseau et potentiellement activer une réaction en chaîne d'influence entraînée par le bouche-à-oreille, de telle sorte que, avec un coût de commercialisation très faible, on peut effectivement espérer toucher une grande partie du réseau. Dans la perspective de la fouille de données, il s'agit d'un problème de maximisation de l'influence (*influence maximization problem*), qui consiste à sélectionner un ensemble des nœuds qui peuvent influencer le plus grand nombre de nœuds dans le réseau social. Le travail de Domingos et Richardson [DR02] est le premier qui considère le marketing viral dans cette perspective.

Autre application des modèles de l'influence, plus intéressante pour nous, est la prédiction des churns. Citons ici le travail de Gupta et al. [DSV+08]. Dans ce travail, les modèles de l'influence sont utilisés de la même manière qu'ils sont employés pour le marketing viral (si on considère que le churn est aussi un phénomène de viralité). On peut utiliser la même idée pour prédire d'autres types d'acte commercial des clients (achats, changement d'offre, etc.). Cette approche permet d'exploiter les données sociales pour prédire des actes commerciaux des clients. L'inconvénient de cette approche, dans le contexte de la thèse, est qu'elle ne prend pas en compte les données tabulaires dans sa modélisation.

2.2.3 Le monitoring des média sociaux

Le *monitoring des média sociaux* désigne les outils permettant de surveiller et d'écouter les conversations dans les média sociaux ou plus largement, sur le Web. Les entreprises utilisent ces outils pour suivre qui parlent d'eux et de leurs concurrents, quand et de quoi ils parlent. Il est aussi un composant important de l'*e-reputation management*, une pratique importante du Social CRM qui consiste à gérer la réputation d'une entreprise ou marque sur Internet.

Parmi les techniques du monitoring des média sociaux, la fouille d'opinion (*opinion mining* est parmi les approches les plus utilisées. La fouille d'opinion, un sous domaine de la fouille de texte, a un grand potentiel d'applications dans plusieurs domaines, tel que montré dans [PL08, GL09]. Les premiers travaux de la fouille d'opinion se sont concentrés sur l'extraction d'opinion à partir des *reviews* des consommateurs sur les produits. Le travail de Dave et al. [DLP03], où apparaît le terme *opinion mining* pour la première fois, consiste à traiter un ensemble de résultats de recherche pour un produit donné, générer une liste des attributs du produit (qualité, fonctionnalités, etc.) et à résumer des opinions sur chacun d'entre eux (négatif, neutre ou positif). Dans [HL04, PE05], la même problématique a été abordée avec les différentes solutions en utilisant des techniques de traitement de langage naturel et d'apprentissage automatique. Dans [LP09] on peut trouver plusieurs techniques de classification d'opinion de *feedback* de la clientèle et de commentaires en ligne. Le microblog Twitter a attiré une grande attention des chercheurs de la fouille d'opinion dans les années récentes [PP10, GHB09, AXV+11, WSLC12, MRMCMVUnL14].

Un autre type de monitoring de média sociaux est la détection (et la prédiction) des tendances dans les média sociaux. Ces outils offrent des indications précieuses permettant d'avoir une vision synthétique des sujets débattus par les internautes en ligne.

La détection des tendances (*emerging trending topic*) est un domaine de recherche qui a suscité l'intérêt pour les applications de text mining depuis une longue période [KGP+04]. Les systèmes de détection de tendances prennent en entrée une grande collecte de données textuelles et identifient les thèmes qui sont inédits ou prennent une importance croissante au sein du corpus. Ces systèmes se sont basés principalement sur les techniques du traitement de langage naturel (e.g extraction de terme à partir de texte) et de l'apprentissage automatique.

La surveillance des tendances sur les média sociaux, notamment Twitter, a déjà fait l'objet d'attention des chercheurs et des professionnels, ce qui entraîne de nombreux algorithmes modifiés et nouveaux pour la recherche d'information ainsi que des outils commerciaux en ligne. Par exemple, *Realtime Twitter Trend* [1] est une application qui suit les mots, ou les sujets, à mesure qu'ils deviennent plus ou moins populaires dans les *tweets* globaux, et ne montre que les plus importants. *Realtime Twitter Trend* présente les tendances en fonction des

1. http ://trendsmap.com/

régions géographiques en temps réel. Dans la recherche scientifique, on peut citer [MK10, CM11, CYS12, GKKL13], dans lesquels on essaye d'extraire et de prédire des tendances (*hot topic*) à partir des posts dans les média sociaux comme Twitter, Facebook.

Avec l'émergence des média sociaux, le monitoring du média sociaux est devenu un composant indispensable dans le processus du Social CRM. Ces techniques permettent d'exploiter des textes, une partie très importante des données de média sociaux. Ces techniques concernent les analyses de comportements collectifs des consommateurs sur les média sociaux. Parmi ces techniques, nous pensons que la fouille d'opinion est intéressante pour l'objectif de la thèse. Les opinions des clients exprimées sur les média sociaux, vis-à-vis de l'entreprise, sont peut-être des indicateurs intéressants pour caractériser l'engagement des clients. Pour prédire les comportements futurs des clients, nous pouvons alors construire des variables explicatives à partir de la fouille d'opinion. Cette idée retombe dans l'approche nous avons mentionnée dans la section précédente : utiliser la classification supervisée pour prédire les comportements des clients. Dans cette thèse, en raison de contraintes de temps, nous n'avons pas implémenté les techniques de fouille d'opinion. Nous gardons cette idée pour de futurs travaux.

2.3 Apprentissage statistique relationnel

La majorité des techniques d'apprentissage automatique classique (le clustering, la classification, la régression - cf. Section 2.1) sont conçues pour modéliser les *données propositionnelles* - c'est à dire les données qui se représentent sous forme de paires attribut-valeur. Ces données sont stockées dans les tables où chaque enregistrement (ligne) correspond à un individu et chaque colonne correspond à un attribut. Du point de vue statistique, les techniques classiques de la fouille de données se sont basées sur l'hypothèse *i.i.d* (*indentical independent distributed* - les valeurs des attributs des différents individus sont indépendantes et ont la même distribution.

D'autre part, les techniques de la fouille de graphe (e.g analyse des réseaux sociaux, les centralités, les modèles de l'influence, cf.Section 2.2) se concentrent seulement sur les données de type relation. Les données de type contenu sur les individu (tabulaires) ne sont pas intégrées dans ces techniques, même si ceci constitue un domaine de recherche émergent.

Les données intercanales que nous traitons dans cette thèse contiennent les deux éléments : les contenus sur les individus (données clientèle, mais aussi interactions textuelles) ainsi que les *relations* entre les individus. Ce type de données est souvent appelé *données relationnelles*. Les données issues des média sociaux sont des données relationnelles : ces données contiennent à la fois les attributs sur les acteurs (e.g leurs profils, les contenus textuels générés par les acteurs) et de relations (d'amitié ou interactions) entre eux.

2.3.1 Introduction à l'apprentissage statistique relationnel

Dans les données relationnelles, il existe non seulement les attributs sur les instances mais aussi les relations *entre les individus*. Les valeurs des attributs d'un individu peuvent dépendre de ceux de ces voisins dans le graphe de relations. L'hypothèse *i.i.d* est donc inappropriée dans ce cas. Les travaux de Jensen et al. [JN02] ont montré que les relations entre les individus dans une base de données relationnelles peuvent avoir un effet significatif sur les comportements des algorithmes d'apprentissage. Pour expliquer les dépendances statistiques entre les individus *connectés*, ils ont introduit les notions de « *lien concentré* » (*concentrated linkage*) et « *d'auto-corrélation relationnelle* »(*relational autocorrelation*). Le lien concentré se produit lorsque plusieurs objets (individus) sont liés à un voisin commun, et l'auto-corrélation relationnelle se produit lorsque les valeurs d'un attribut sont très uniformes parmi des objets ayant un voisin commun. Ils ont constaté que le lien concentré et l'auto-corrélation relationnelle sont des caractéristiques des données relationnelles. Ils ont aussi défini des mesures pour quantifier ces caractéristiques.

D'autres exemples de dépendances entre les individus liées dans les données relationnelles sont énoncés dans le domaine d'analyse de réseaux sociaux [WF94]. Dans les réseaux sociaux, il existe deux caractéristiques très connues : l'*homophilie*(*homophily*) et l'*équivalence stochastique* (*stochastic equivalence*).

- L'homophilie est une caractéristique d'un réseau social, selon laquelle les relations entre les nœuds présentant des caractéristiques similaires sont plus fortes que les relations entre les nœuds ayant des caractéristiques différentes. L'homophilie fournit une explication aux modèles de données apparaissant souvent dans les réseaux sociaux, comme la transitivité (« l'ami d'un ami est un ami ») et l'existence de sous-groupes de nœuds homogènes (e.g communauté) ou encore le phénomène de l'influence dans les réseaux sociaux.
- L'équivalence stochastique est aussi une caractéristique souvent observée dans les réseaux sociaux. Dans un réseau ayant cette caractéristique, les nœuds peuvent être divisés en groupes, tels que les membres d'un même groupe ont des profils similaires de relations. Par exemple, les pages web peuvent être catégorisées en deux groupes : *autorités* et *hubs* (*authority and hub* [Kle99]). Les *autorités* ont beaucoup de liens en provenance des *hubs* et les *hubs* ont beaucoup de liens sortants vers les autorités. Les thèmes des pages web *autorités* sont généralement fortement auto-corrélés lorsque ces pages sont liées par des *hubs* communs.

Les techniques d'*apprentissage statistique relationnel* [GT07] ont été conçues pour faire face aux données relationnelles, particulièrement pour exploiter les caractéristiques de ces données (e.g auto corrélation, homophilie, etc.) pour en extraire des informations utiles. L'apprentissage statistique relationnel est un domaine de recherche très actif. Il concerne

plusieurs sous-domaines de l'apprentissage automatique et de la fouille de données. Dans la communauté de la fouille de données, l'apprentissage statistique relationnel est appelé « fouille de données multi-relationnelle »ou « fouille de données multi-table »(*multi-relational data mining*) ([Dom03, Dŏ3]). Dans la communauté de la *Programmation Logique Inductive* (*Inductive Logic Programming*), l'apprentissage statistique relationnel est souvent considéré comme une combinaison de l'*apprentissage relationnel* (avec la programmation logique du premier ordre) et l'apprentissage statistique (i.e traiter des données incertaines) ([GT07]) ; il est aussi appelé *apprentissage logique probabiliste* (*Probabilistic Logic Learning* [RK03]). Les techniques d'apprentissage statistique relationnel utilisent souvent les *modèles graphiques relationnels* pour modéliser la distribution de données. Les modèles graphiques relationnels sont des extensions des *modèles graphiques probabilistes* [KF07, Hec08] comme les réseaux Bayésiens ou de Markov avec l'intégration de structures riches de données relationnelles.

Dans la suite nous citons quelques travaux typiques concernant les tâches populaires d'apprentissage statistique relationnel : la classification collective, le clustering basé sur les liens (ou clustering de graphe) et la prédiction de lien.

2.3.2 Classification collective

Comme dans l'apprentissage traditionnel, la classification concerne le problème de la prédiction des étiquettes (i.e groupe prédéfini) des objets/individus dans les données. En apprentissage statistique relationnel, la classification utilise les attributs observés, les étiquettes observées et le graphe de relations entre les individus pour inférer les étiquettes non-observées. La classification en apprentissage statistique relationnel est souvent appelée *classification collective* [JNG04] ou *classification intra-réseau* (*within network classification* [DK09]) pour souligner le fait que les relations entre les objets doivent être prises en compte afin d'améliorer la précision de prédiction.

Une approche très connue de la classification collective est l'*inférence collective* [JNG04, MP07]. Dans l'apprentissage, les données tabulaires sont utilisées pour apprendre un modèle par apprentissage supervisé. L'inférence collective est effectué lors du déploiement du modèle. La procédure d'inférence collective consiste à effectuer des jugements statistiques sur les étiquettes pour un ensemble d'instances de données de test de manière simultanée. L'objectif de l'inférence collective est d'utiliser les *relations* entre les instances, de profiter de l'auto-corrélation dans les données pour améliorer la précision de prédiction. L'inférence collective est le contraire de l'*inférence individuelle*, selon laquelle on calcule à chaque fois l'étiquette d'une seule instance dans l'ensemble de test. Les méthodes supervisées conventionnelles n'utilisent que l'inférence individuelle (toutes les instances sont indépendantes). L'inférence collective est basée sur l'*assomption de Markov* : l'étiquette d'un nœud ne dépend que de celles de ses voisins dans le graphe. L'inférence collective a été implémentée dans un

outil open-source sous le nom *NetKitSRL* [1]. Dans cet outil on peut trouver plusieurs implémentations d'inférence collective pour la tâche de classification collective [MP07].

Une autre approche de la classification collective concerne les modèles graphiques probabilistes relationnels. Par exemple, Taskar et al. [TSK01] ont utilisé une extension relationnelle des réseaux Bayésiens pour la tâche de classification dans une base de données relationnelles concernant les articles scientifiques. Dans [TAK02], ils ont utilisé une extension relationnelle des réseaux de Markov pour la classification des pages Web à partir des contenus des pages et des *hyperliens* entre les pages. Les expérimentations sur les données réelles montrent que l'utilisation des *relations* dans les données par un modèle probabiliste relationnel peut améliorer significativement la précision de classification par comparaison avec les modèles propositionnels (e.g la régression logistique) qui n'utilisent que les données en forme attribut-valeur.

La classification collective combine les données attribut-valeur et le graphe social pour inférer les étiquettes sur les individus. Les techniques de la classification collective sont donc applicables sur les données intercanales que nous avons dans cette thèse. Pourtant, le problème résolu par la classification collective est différent de la problématique posée dans la thèse. La classification collective consiste à prédire les étiquettes non-connues dans un jeu de données relationnelles non-datées, dans lequel on connaît les étiquettes sur certains individus. La problématique de la thèse consiste à prédire les actes futurs des clients. Dans la classification collective, on effectue l'apprentissage et l'inférence sur un seul jeu de données relationnelles. On ne peut pas utiliser la même démarche décrite dans la figure 2.1 dans laquelle on a deux jeux de données à deux instants, un pour l'apprentissage et l'autre pour le déploiement. L'application de la classification collective pour l'objectif de prédiction de la thèse n'est donc pas évidente.

2.3.3 Clustering basé sur les liens

La tâche de clustering dans un réseau social est souvent appelé la détection de communauté : il s'agit d'identifier des sous-groupes ou communautés d'acteurs dans le réseau. Une communauté est souvent définie comme un groupe d'acteurs avec de fréquentes interactions survenant entre eux. Ces fréquentes interactions se matérialisent par des zones denses de relations dans le graphe. La détection de communauté peut être utilisée pour une analyse plus approfondie comme la visualisation, le marketing viral, déterminer les facteurs de causalité de la formation du groupe, la détection de l'évolution du groupe ou des groupes stables. Les communautés sont souvent les groupes d'amis ou les gens ayant les mêmes intérêts (le caractère d'homophilie), la détection de communauté est donc utile pour analyser

1. http ://www-bcf.usc.edu/~macskass/NetKit-desc.html

ou prédire les comportements des acteurs dans un réseau social.

Différentes approches ont été proposées pour la détection de communautés. L'approche la plus connue est la maximisation de modularité [New06, BGLL08]. Cette approche consiste à calculer le partitionnement optimal du graphe en maximisant la *modularité*, une mesure pour la qualité d'un partitionnement des nœuds d'un graphe. Les autres approches sont les modèles d'espace latent [HRH02] (transformer le graphe vers un espace latent de faible dimension de telle sorte que la distance ou la similarité entre les nœuds sont maintenus), les modèles de bloc [NS01, KN11] (supposer que les nœuds du graphe appartiennent à des blocs non observés qui décrivent leur connectivité aux autres nœuds), le clustering spectral [Lux07] (calculer quelques valeurs propres et les vecteurs propres correspondants de la matrice *Laplacienne* du graphe). Récemment, il y a aussi des efforts d'utiliser conjointement le graphe et le contenu (i.e les attributs sur les nœuds) pour la détection de communauté [RFP12, CBP13]. Dans ces travaux, on combine la mesure de similarité basée sur le contenu et la densité de liens pour le partitionnement des nœuds du graphe. Combe et al. [CLEZG12] ont étudié différentes techniques de clustering basées sur une combinaison de deux types d'information : le contenu (données tabulaires) et le graphe (*hybrid clustering*). Ils ont fait valoir que, selon le type de données que nous avons et le type de résultats que nous voulons, le choix de la méthode de *clustering* (comment combiner les deux types d'information) est important.

Le clustering basé sur les liens est étroitement liée aux enjeux de cette thèse. Il permet d'exploiter des informations de type « relation » dans les données intercanales et parfois d'exploiter conjointement relations et contenus. Pourtant, le problème attaqué par les techniques de clustering (chercher les groupes homogènes d'individus) n'est pas en lien direct avec la problématique principale de la thèse (prédire les comportements des clients). Ces techniques sont intéressantes si on les utilise comme un prétraitment pour la classification supervisée, comme décrite ci-après dans la section 2.3.5.

2.3.4 Prédiction de liens

La prédiction des liens consiste à déterminer si une relation existe entre deux instances à partir des attributs des instances et des relations entre eux. La prédiction de liens a plusieurs applications. Par exemple, il peut être nécessaire de trouver des liens manquants qui ne sont pas présents dans les données en raison d'une collecte de données incomplète. De même, nous pourrions être intéressés à prédire les liens cachés, où l'on suppose qu'il existe des interactions, mais qui sont non observables pour des raison de confidentialité par exemple, et l'objectif est de découvrir et modéliser ces interactions. Sinon, nous pouvons chercher à prédire des liens futurs dans une évolution du réseau, tels que de nouvelles amitiés ou des connexions qui seront formées prochainement. Cette dernière application régit le principe

de la recommandation d'amis dans les réseaux sociaux en ligne.

Plusieurs techniques ont été proposées pour la prédiction de liens dans différents domaines. Popescul et al. [PU03] ont introduit la *Régression Logistique Structurelle*, une extension de la régression logistique pour exploiter la structure relationnelle de données. Ils ont utilisé la méthode pour construire des modèles de prédiction de liens pour la tâche de prédire les citations dans la littérature scientifique en utilisant des données relationnelles collectées à partir du moteur de recherche de CiteSeer [1]. Huang [Hua06] a proposé une méthode de prédiction de liens basée sur la topologie du graphe. L'idée est d'utiliser les mesures de la topologie du graphe (e.g le coefficient de clustering [WF94, p. 243]) comme des statistiques décrivant les occurrences de liens dans les graphes ; à partir de ces statistiques on peut construire des modèles de probabilité pour prédire de nouveaux liens. Ils ont utilisé ces modèles pour prédire des liens dans les graphes sociaux réels y compris un graphe issu de Facebook. La classification supervisée peut également être utilisée pour la prédiction de liens [HCSZ06]. Une instance de données pour la classification correspond à une paire de nœuds du graphe, l'étiquette sur une paire de nœuds signifie s'il existe un lien entre ces nœuds. Le défi majeur de cette approche est de choisir un ensemble de variables explicatives sur les paires de nœuds pour la tâche de classification. Dans [HCSZ06, LK07], on peut trouver plusieurs choix possibles : les variables basées sur les voisins communs, les variables basées sur les chemins entre deux nœuds, les attributs sur les nœuds et sur les liens etc. Zheleva et al. [ZGGK09] ont utilisé cette approche pour la prédiction de liens dans les réseaux sociaux multi-relationnels (i.e plusieurs types de liens). En plus des approches citées au-dessus, les modèles graphiques relationnels (les réseaux Bayésiens relationnels et les réseaux de Markov relationnels) sont aussi souvent utilisés pour la prédiction de liens [TWAK03, TA07].

Dans le contexte de la thèse, la prédiction de liens peut être appliquée sur les données intercanales pour certaines applications. On peut par exemple prédire l'établissement de liens sociaux entre les clients dans les média sociaux ou prédire des paires de clients « similaires ». Ces applications sont intéressantes dans certaines situations (e.g pour un moteur de recommandation d'amis dans un forum d'entraide de la marque), mais elles ne sont pas nos premières préoccupations dans ce travail.

2.3.5 Dimensions sociales - combinaison du graphe social et du contenu pour un apprentissage supervisé

Certains travaux ont utilisé le clustering basé sur les liens (clustering de graphe) pour l'apprentissage supervisé. Citons ici le travail de Tang et al. [TL11] qui ont utilisé le cluste-

1. http ://citeseer.org/

ring de graphe pour la tâche de classification des nœuds dans un graphe social. L'idée de cette méthode est de transformer le graphe social en caractéristiques des nœuds en utilisant un algorithme de clustering basé sur les liens. Les appartenances des nœuds aux différents *cluster* de graphe social (les communautés) sont utilisées comme variables explicatives pour la tâche d'apprentissage supervisé. L'appartenance des nœuds à une communauté est appelé une *dimension sociale* (*social dimension* en anglais). Le terme *dimension sociale* a été introduit par Tang et al. [TL11], d'où vient l'approche d'apprentissage relationnel basée sur les dimensions latentes.

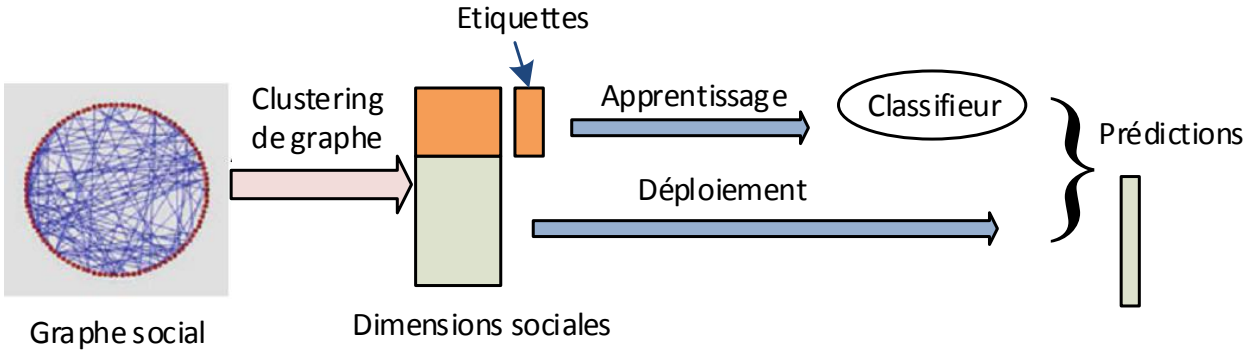


FIGURE 2.2 – L'approche basée sur les dimensions sociales [TL11] pour prédire des étiquettes non-connues dans un graphe partiellement étiqueté

En utilisant les dimensions latentes, cette approche permet d'exploiter le graphe social à l'échelle globale et pas seulement le voisinage d'un nœud comme l'approche d'inférence collective ou les modèles de l'influence. L'intuition derrière cette approche est de détecter des groupes des nœuds dans le graphe social et de déduire les étiquettes non-connues sur les nœuds en se basant sur l'hypothèse suivante : les nœuds dans le même groupe ont tendance à avoir des étiquettes similaires (même se ils ne sont pas directement connectés). Ainsi, les dimensions latentes sont utilisées comme variables explicatives pour l'apprentissage supervisé, pour déduire les étiquettes non-connues dans un graphe partiellement étiqueté (figure 2.2). N'importe quel algorithme de clustering peut être utilisé pour extraire les dimensions sociales, mais le *clustering spectral* [Lux07] a été démontré expérimentalement meilleur par [TL11]. Avec le clustering spectral, une dimension sociale est un nombre réel représentant l'appartenance des nœuds à un groupe particulier.

Tang et al. [TL11] ont mené des expérimentations sur des jeux de données réelles de médias sociaux (recueillies à partir de BlogCatelog et Flickr). Le problème est de prédire les groupes d'intérêt des utilisateurs sur ces média (sur ces médias les utilisateurs se souscrivent à des différents groupes d'intérêt). Ils ont montré que leur approche basée sur les dimensions latentes est meilleure que l'inférence collective [MP07].

Un avantage de l'approche de Tang et al. [TL11] est la capacité de combiner les données

tabulaires et le graphe social. Il suffit d'utiliser conjointement les dimensions sociales avec les données tabulaires sur les individus (le contenu) pour l'apprentissage supervisé.

L'idée proposée dans Tang et al. [TL11] est intéressante pour la thèse. Cette approche permet de combiner les interactions sociales issues des média sociaux (via les dimensions sociales) avec les données tabulaires (données clientèle). En d'autres termes, les dimensions sociales peuvent être utilisées comme variables explicatives conjointement avec les autres types de variables pour l'apprentissage supervisé. L'inconvénient de cette approche, dans le contexte de la thèse, est le même que celui de la classification collective : elle est conçue pour prédire les étiquettes non-connues dans un jeu de données relationnelles non-datées, dans lequel on connaît les étiquettes sur certains individus. Nous pouvons quand même adapter cette approche pour prédire des comportements futurs des clients comme illustrée dans la figure 2.1. Ici, on construit un graphe social à chaque instant, et ensuite extrait les dimensions sociales à partir du graphe à chaque instant. Lors de l'apprentissage, les dimensions sociales extraites à instant $t0$ sont utilisées comme variables explicatives. Lors de déploiement, on utilise les dimensions sociales extraites à instant $t1$. On peut espérer que les dimensions sociales extraites à deux instants signifient les mêmes caractéristiques latentes sur les individus. Le problème ici est que, si le graphe social évolue beaucoup entre ces deux instants, la structure de clusters de graphe pourrait changer. Les dimensions latentes extraites à deux instants ne porteraient pas les mêmes sémantiques. Dans ces cas, la performance de prédiction pourrait se dégrader (nous verrons dans la section 3.7.2.1). Dans cette thèse, nous utiliserons quand même cette approche comme une méthode de référence.

2.4 Conclusion

Dans ce chapitre, nous avons décrit différentes applications des techniques de la fouille de données dans le CRM et le Social CRM. Ce sont des techniques de l'apprentissage automatique traditionnelles (le clustering, la classification, la régression), les techniques de la fouille de média sociaux et les techniques de l'apprentissage statistique relationnel. Presque toutes les techniques existantes de la fouille de données pour le CRM sont *mono-canal* : chacune des techniques que nous avons répertoriées traite un type particulier de données issues d'un seul canal de communication entre les clients et l'entreprise (soit des données clientèle de l'entreprise, soit les média sociaux). Nous avons identifié quelques techniques applicables pour la thèse, mais chacune de ces techniques a souvent ses propres inconvénients.

Les techniques classiques de la fouille de données et de l'apprentissage automatique sont adaptées aux données tabulaires. Ce sont les techniques de l'apprentissage supervisé (la classification la régression) et le clustering avec les données clientèle (concernant des

transactions et les données de facturation). Les techniques de l'apprentissage supervisé sont particulièrement intéressantes pour la thèse, car pour le CRM et pour l'engagement avec les clients, elles permettent de prédire les comportements commerciaux des clients, de prédire leur la valeur potentielle (peut être considérée comme une mesure de l'engagement). La limitation principale de ces techniques est qu'elles ne sont utilisées que pour les données de type attribut-valeur. Dans le CRM classique, les données clientèle sont d'abord transformées en données attribut-valeur avant d'appliquer ces techniques. Dans le cadre d'une stratégie intercanale, particulièrement avec la présence des données relationnelles issues de média sociaux, la transformation de données en attribut-valeur (la construction des variables explicatives) est non intuitive.

Les techniques de la fouille de média sociaux sont adaptées aux données sous la forme de graphes. Les applications de ces techniques en CRM sont variées : le marketing viral, l'identification des clients « importants » ou le monitoring de média sociaux. Ces applications ne sont pas pertinentes en tant que telles pour l'objectif de la thèse. Pourtant, certaines de ces approches peuvent être utilisée. Les mesures de l'influence peuvent être utilisées comme variables explicatives dans l'apprentissage supervisé. Ainsi, les modèles de l'influence peuvent être utilisé pour prédire des actes commerciaux des clients. La limitation des modèles de l'influence est qu'ils ne sont pas capables de prendre en compte les données de type contenu sur les individus, même si quelques travaux commencent à apparaître.

Les techniques d'apprentissage statistique relationnel sont capables d'exploiter simultanément les attributs et les relations dans les données. La classification collective est particulièrement intéressante pour notre domaine applicatif. Ces techniques permettent de prédire des étiquettes sur les individus dans un jeu de données relationnelles. La limitation des techniques de classification collective est qu'elles sont applicables sur un jeu de données statiques. Ces techniques ont pour but de déduire les étiquettes non-connues dans un graphe partiellement étiqueté. Dans la thèse, nous avons besoin de techniques permettant de prédire des événements dans le futur.

Le clustering basé sur les liens, utilisé comme prétraitement des données relationnelles, peut être utilisé dans le contexte de la thèse (approche de Tang et al. [TL11]). L'idée est de considérer les appartenances des individus aux clusters du graphe social comme des variables explicatives pour l'apprentissage supervisé. Nous pouvons adapter cette approche pour la problématique de la thèse (cf. section 2.3.5). Nous verrons dans la suite que cette méthode est utilisée comme une méthode de référence.

Les techniques analysées dans ce chapitre ne permettent pas de capter la dynamique de données, dans le sens où les caractéristiques des données peuvent évoluer au fil du temps. Ces techniques ne prennent pas en compte l'apparition de nouveaux types de contenus ou l'évolution des dépendances statistiques entre différents éléments des données.

Dans la suite (Chapitre 3 et 4) nous retrouvons certaines techniques listées dans cette partie (notamment les méthodes basées sur l'apprentissage supervisé, les modèles de l'influence sociale et l'approche de Tang et al. [TL11]) comme certaines des méthodes de référence que nous utilisons pour comparer avec la méthode que nous proposons.

Dans cette thèse, nous allons concevoir une nouvelle méthode pour exploiter conjointement des données tabulaires et des données relationnelles. Il s'agit d'une méthode de fouille de données relationnelles. Elle tient compte de la dynamique des données, elle est capable d'apprentissage incrémental, c'est-à-dire qu'elle permet d'intégrer des nouvelles données à chaque pas de temps pour mettre à jour le modèle, au lieu de recommencer l'apprentissage à partir de zéro.

CHAPITRE 3

APPRENTISSAGE INCRÉMENTAL AVEC UN MODÈLE À FACTEURS LATENTS

Sommaire

Dans ce chapitre nous présentons notre méthode pour répondre à la problématique principale de la thèse : utiliser les données issues de plusieurs canaux (y compris et particulièrement les média sociaux) pour prédire des comportements des clients. Nous nous intéressons à exploiter les données intercanales qui contiennent à la fois les interactions sociales, les contenus créés par les individus dans les média sociaux et les données tabulaires (issues de la base de données clientèle de l'entreprise). Nous proposons d'utiliser un *modèle à facteurs latents* qui consiste à caractériser les individus par un ensemble de variables latentes. De plus, pour adapter au fait que les données issues de média sociaux sont datées et à la dynamique de ces données, nous proposons d'utiliser un apprentissage incrémental pour mettre à jour les variables latentes à chaque pas de temps. La méthode proposée est s'appelle *Apprentissage Incrémental des Modèles à Facteurs Latents (AIMFL)*. Dans les deux premières sections, nous révisons brièvement les travaux connexes : les modèles à facteurs latents et les méthodes d'apprentissage incrémental. Nous présentons ensuite notre algorithme d'apprentissage des facteurs latents. Enfin nous présentons les premières expérimentations sur les données synthétiques pour illustrer le fonctionnement de notre méthode.

3.1 Les modèles à facteurs latents

Notre approche d'apprentissage est inspirée *des modèles à facteurs latents* [BKM11]. Les modèles à facteurs latents (MFL), également appelés *modèles à variables latentes*, sont utilisés en statistique et en apprentissage automatique depuis longtemps. Un MFL est un modèle statistique qui représente chaque instance de données (individu) par un ensemble de variables latentes. Comme les variables observées, les variables latentes peuvent être continues (dans l'analyse de facteur latent) ou catégorique (dans l'analyse de classe latente). Dans ce travail, nous nous intéressons aux modèles avec les variables latentes continues de valeur réelle.

L'hypothèse de base d'un modèle à facteurs latents est que les variables latentes d'un individu manifestent toutes les *observations* sur cet individu. Dans notre cas, les observations sont les attributs sur les individus ainsi que ses relations/interactions avec les autres dans les cas de données relationnelles.

3.1.1 Modèles à facteurs latents pour les données attribut-valeur

Les modèles à variables latentes sont souvent utilisés pour transformer ou réduire le nombre de dimensions de données. Un grand nombre de variables observables sont projetées dans un espace latent de faible dimension, et sont ainsi agrégées en un modèle représentant un concept sous-jacent, ce qui rend plus facile la compréhension des données. Parmi les

méthodes d'extraction des facteurs latents, l'*analyse en composantes principales* (ACP) [Shl05] est la plus populaire. L'ACP est une méthode statistique qui utilise une transformation orthogonale pour convertir un ensemble de variables observées (éventuellement corrélées) en un ensemble de variables latentes non corrélées (appelées les composantes principales). Les composantes principales correspondent aux dimensions (dans l'espace observable des données) dont les variances sont les plus grandes. Pour trouver les composantes principales, une méthode souvent utilisée est de calculer les premières valeurs propres et les vecteurs propres correspondants de la matrice de covariance.

3.1.2 La factorisation de matrice

La *factorisation de matrice* (FM) peut être considérée comme une méthode de modélisation à facteurs latents. La FM consiste à décomposer une matrice de données à grande dimension (la matrice dans laquelle les lignes correspondent à des instances et les colonnes correspondent à des variables) en matrices à dimension plus faible. Chaque dimension de la matrice à dimension inférieure correspond à une variable latente. Une méthode très connue de FM, l'*analyse sémantique latente* [DDF$^+$90] a été utilisée depuis longtemps dans la fouille de texte. L'analyse sémantique latente utilise la *décomposition en valeurs singulières* pour décomposer la matrice des occurrences, par exemple la matrice qui représente les occurrences des termes dans chaque document. Selon cette décomposition, les documents et les termes sont projetés dans un espace de « concepts » à faible nombre de dimensions.

Récemment, la FM a été appliquée au filtrage collaboratif avec grand succès [KBV09]. Le filtrage collaboratif est une approche populaire des systèmes de recommandation, dont le but est de recommander des articles (e.g films, produits à acheter) susceptibles d'intéresser un utilisateur. Le filtrage collaboratif permet de faire des prévisions automatiques sur les intérêts d'un utilisateur en collectant des préférences ou des informations de goût de nombreux utilisateurs. L'idée est d'utiliser les opinions et évaluations d'un groupe pour faire des recommandations pour un individu. La préférence ou les goûts d'un utilisateur pour un article est souvent exprimée par une note d'évaluation. Les données en entrée (les données d'apprentissage) d'un système à filtrage collaboratif est une *matrice d'usage (rating matrix)* $\mathbf{R}$ de taille $n \times m$, où n est le nombre d'utilisateurs et m est le nombre d'articles. L'élément r_{ij} de la matrice d'usage R est le note que l'utilisateur i a donnée pour l'article j, ce qui représente les préférences de l'utilisateur pour l'article. Les techniques de FM aident à décomposer la matrice d'usage R en 2 matrices de rang inférieur :

$$\mathbf{R} \approx \mathbf{U}\mathbf{P}^T \tag{3.1}$$

où $\mathbf{U}$ et $\mathbf{P}$ sont des matrices de taille $n \times d$ et $m \times d$ respectivement. d est le nombre de

facteurs latents, qui est inférieur à n et m. Les lignes des matrices **U** et **P** représentent les facteurs latents des utilisateurs et des articles. Si l'on note la ligne i de **U** par u_i et la ligne j de **P** par p_j, la décomposition de la matrice (équation 3.1) peut être réécrite comme suit :

$$r_{ij} \approx u_i p_j^T \tag{3.2}$$

La décomposition de la matrice ci-dessus est interprétée comme ceci : le vecteur u_i représente d caractéristiques latentes de l'utilisateur i et le vecteur p_j représente d caractéristiques latentes de l'article j. Le produit scalaire de ces deux vecteurs $u_i p_j^T$ capte l'interaction entre l'utilisateur i et l'article j - il représente donc l'intérêt de l'utilisateur par l'article.

La matrice **R** contient à la fois les valeurs connues (notes de l'ensemble de l'apprentissage) et les valeurs manquantes (notes inconnues que nous essayons de prévoir). Les facteurs latents des utilisateurs et des articles peuvent être calculés à partir de l'ensemble d'apprentissage en utilisant la méthode des moindres carrés :

$$\mathbf{U}^\star,\ \mathbf{P}^\star = \arg\min_{\mathbf{U},\mathbf{P}} \frac{1}{2} \sum_{(i,j)\in\mathcal{K}} \left(r_{ij} - u_i p_j^T\right)^2 + \frac{\lambda}{2}\left(\|u_i\|^2 + \|p_i\|^2\right) \tag{3.3}$$

où $\mathcal{K}$ est l'ensemble de paires utilisateur-article pour lesquelles la note est connue et λ est un paramètre de régularisation.

À la différence des facteurs latents calculés avec l'analyse en composantes principales ou l'analyse sémantique latente (ou plus précisément, la décomposition en valeurs singulières), les facteurs latents définis dans l'équation 3.1 ne sont pas indépendants ; on ne peut pas garantir que les lignes des matrices **U** et **P** soient orthogonales. On peut considérer la décomposition 3.1 comme une approximation de la décomposition en valeurs singulières, où la contrainte d'orthogonalisation est oubliée. L'avantage de cette décomposition est qu'elle est beaucoup moins coûteuse (en calcul) et robuste au passage à grande échelle. C'est pour cela qu'elle a été utilisée dans les problèmes de filtrage collaboratif de très grande taille.

La FM a été adaptée pour prendre en compte la dynamique de données. Pour ce faire, les facteurs latents sont considérés comme des fonctions qui changent au fil du temps. Par exemple, Koren et al. [KBV09] a utilisé la version dynamique de la FM pour le filtrage collaboratif. Ce modèle dynamique est motivé par le fait que les *notes de préférences* sont horodatées et que les goûts d'un utilisateur peuvent changer au fil du temps. Les résultats des expérimentations (pour la tâche de recommandation) ont montré que ce modèle dynamique donne une performance meilleure que la factorisation statique de matrice.

3.1.3 Modèles à facteurs latents pour l'apprentissage statistique relationnel

Les techniques de FM ont été adaptées pour modéliser les données relationnelles. Les données relationnelles sont représentées par plusieurs matrices - en plus de la matrice d'attribut-valeur qui représente les attributs des individus, il y a d'autres matrices qui représentent les relations ou interactions entre les individus.

Singh et Gordon [SG08] ont introduit la *factorisation collective de matrices* (FCM) qui permet de décomposer simultanément plusieurs matrices. Les techniques de FCM ont été conçues pour les données relationnelles où il y a plusieurs types d'objets et plusieurs types de relations entre les objets. Chaque relation est représentée par une *matrice relationnelle*. Par exemple, dans le domaine de la recommandation de films, nous avons comme type d'objets : les utilisateurs, les films et les genres (d'un film). Les matrices de relations sont : la matrice d'usage utilisateur-film (quels utilisateurs apprécient quels films), la matrice des genres des films (quels films appartient à quels genres). On peut utiliser la FM classique pour décomposer ces deux matrices séparément, mais cette approche ne peut pas exploiter les corrélations entre différents types de relations. La FCM a été proposée pour remédier à ce problème en décomposant simultanément plusieurs matrices relationnelles. L'idée est de partager les paramètres latents d'un objet sur plusieurs décompositions lorsqu'il participe à plusieurs relations.

Li et Yeung [LY09] ont proposé une autre approche permettant de modéliser les facteurs latents dans un domaine relationnel. La FM classique a été étendue pour prendre en compte à la fois les informations de type *contenu* et les informations de type *relations*. Les données sont représentées par deux matrices : une matrice de contenu qui représente les valeurs des attributs sur les objets et une matrice de relations qui représente les relations/interactions entre les objets, i.e la matrice d'adjacence du graphe social ou du graphe d'interactions. Pour déterminer les facteurs latents des objets, Li et Yeung [LY09] ont introduit la *factorisation régularisée relationnelle de matrice* (FRRM) (*regularized relational matrix factorization*). Cette approche est basée sur la méthode des moindres de carrés comme dans l'équation 3.3 et pour prendre en compte les relations entre les objets (utilisateurs), on ajoute le terme de *régularisation relationnelle* qui est défini comme suit :

$$\mathbf{r} = \frac{\alpha}{2}\sum_i\sum_j \mathbf{S}_{ij} \left\| u_i - u_j \right\|^2 \tag{3.4}$$

où u_i est le vecteur de facteurs latents de l'objet i, $\mathbf{S}$ est la matrice d'adjacence des relations et α est un paramètre du modèle. Ce terme de régularisation permet de rapprocher les objets connectés (par les relations ou interactions) dans l'espace latent. Les facteurs latents appris

sont ensuite utilisés comme les variables explicatives pour une classification supervisée.

Récemment, un modèle à facteur latent avec le terme de régularisation relationnelle a été utilisé dans les réseaux hétérogènes où les nœuds sont de types différents [JDG14]. Le problème est d'étiqueter les nœuds dans ces réseaux (chaque type de nœuds correspond à un ensemble particulier de catégories possibles). L'idée de la méthode proposé dans [JDG14] est de projeter les nœuds dans un espace latent commun à tous les types des nœuds et de supposer que deux nœuds connectés sont proches dans cet espace. Les étiquettes sur les nœuds sont ensuite déduites de leurs positions dans l'espace latent.

Gao et al. [GDG12] ont proposé un modèle à facteurs latents avec le terme de régularisation relationnelle pour la prédiction de liens (i.e prédire l'apparition des nouveaux liens dans des graphes de données dynamiques). Ce modèle permet d'intégrer simultanément différentes types d'informations : la structure topologique du réseau, le contenu des nœuds dans le réseau et l'information de proximité locale des nœuds. L'apprentissage est effectué à base de factorisation en matrices non-négatives (c'est-à-dire, ici les facteurs latents sont les nombres réels non-négative). Les expérimentations sur plusieurs ensembles de données du monde réel ont montré que ce modèle surpasse les méthodes de l'état de l'art en termes de performance de prédiction.

Les modèles à facteurs latents et plus précisément les techniques de factorisation de matrices et leurs extensions constituent une famille des techniques récentes d'apprentissage statistique relationnel. Ces techniques nous intéressent parce qu'elles sont capables d'exploiter simultanément les données relationnelles (sociales) et les données tabulaires. Nous rappelons que cela est la problématique majeure dans la fouille de données intercanales. En plus des travaux présentés ci-dessus, on peut trouver d'autres méthodes à facteurs latents pour l'apprentissage relationnel dans [Li10]. En termes de performance, ces méthodes donnent souvent des résultats comparables avec les méthodes de l'état de l'art avec les jeux de données issues de média sociaux, particulièrement pour la tâche de classification des objets. Nous verrons par la suite qu'il est possible d'étendre ces modèles pour gérer l'aspect dynamique de données, en utilisant une adaptation dynamique des variables latentes.

3.2 Les techniques d'apprentissage incrémental

Les techniques d'apprentissage classiques, y compris les techniques basées sur les variables latentes, ont été conçues pour un apprentissage hors-ligne (ou *batch learning*). L'apprentissage hors-ligne est l'apprentissage d'un modèle sur un jeu de données statique et disponible au moment de l'apprentissage. Ce mode d'apprentissage n'est pas adapté aux cas où (i) les données sont volumineuses et non modifiables une fois chargée en mémoire ou (ii) les données ne sont pas complètes au moment de l'apprentissage car elles arrivent de

manière continue (en flux). Les techniques d'apprentissage incrémentales ont été conçues pour faire face à ces problèmes.

Classiquement, l'apprentissage incrémental est défini sur les données attribut-valeur. Dans [Sal12], un algorithme incrémental est défini de la manière suivante : pour n'importe quel exemple $x_1, x_2, ..., x_n$ il est capable de produire des modèles $f_1, f_2, ..., f_n$ tel que f_{i+1} ne dépende que de f_i et l'exemple courant x_i. La notion « d'exemple courant »peut être étendue à un résumé des derniers exemples vus, résumé utile à l'algorithme d'apprentissage utilisé.

L'apprentissage incrémental concerne à la fois l'apprentissage non supervisé (clustering incrémental) et l'apprentissage supervisé (par exemple la classification incrémentale).

Les critères d'un algorithme d'apprentissage incrémental sont les suivants (selon Domingos et Hultens [DH01] , cité dans [Sal12]) :

- durée faible et constante pour apprendre les exemples
- lecture une seule fois des exemples et dans leur ordre d'arrivée
- utilisation d'une quantité de mémoire fixée a priori
- production d'un modèle proche de celui qui aurait été généré s'il n'y avait pas eu la contrainte de flux
- possibilité d'interroger le modèle à n'importe quel moment
- possibilité de suivre les changements de concept (concept drift). Les changements de concepts sont une caractéristique des données dynamiques, dans lesquelles les propriétés statistiques des variables (e.g. la dépendance statistique entre les variables cibles et les variables explicatives) évoluent au fil du temps.

Dans [SL10, JK12], on peut trouver une liste des techniques d'apprentissage incrémental les plus utilisées. La majorité de ces techniques sont des adaptations de techniques d'apprentissage bien connues. On peut citer les versions incrémentales de l'arbre de décision [UU89], de la machine à vecteurs de support [CP01, DC03], du classifieur Bayésien naïf [LIT92] (naturellement incrémental), de l'approche k-plus proches voisins [DC03, CP01].

Toutes les techniques d'apprentissage incrémental citées ci-dessus concernent seulement les données attribut-valeur. Dans ce travail, nous adoptons le concept d'apprentissage incrémental pour les données relationnelles. Nous nous intéressons particulièrement à la *capacité de mettre à jour un modèle d'apprentissage avec les nouvelles données* (collectées depuis la dernière itération d'apprentissage). La notion « d'exemples courants »dans la définition ci-dessus est ainsi étendue pour indiquer les données que l'algorithme reçoit à chaque incrément pour mettre à jour le modèle f_i et obtenir le modèle f_{i+1}. Ces données contiennent à la fois les éléments attribut-valeur (les valeurs des attributs sur les individus) mais aussi les relations entre les individus.

Ainsi, notre objectif n'est pas de répondre à tous les critères de l'apprentissage incrémental cités ci-dessus. Le critère auquel nous nous intéressons principalement est le deuxième :

lecture une seule fois des exemples dans l'ordre d'arrivée. Dans notre travail, « les exemples courants »sont les données relationnelles représentées par un graphe attribué (décrit dans les paragraphes suivants). Concernant les critères de temps de calcul et d'utilisation de la mémoire, nous ne pouvons pas garantir un temps de calcul et une mémoire fixe, car ils dépendent de la taille des données à chaque incrément. Naturellement, notre approche est adaptée au changement de concept (le dernier critère), parce qu'à chaque incrément, seule la partie récente des données est utilisée pour mettre à jour le modèle.

3.3 Représentation des données

Nous nous intéressons aux données intercanales dans lesquelles il y a les interactions sociales, les contenus créés par les individus dans les média sociaux et les données tabulaires (issues de la base de données clientèles de l'entreprise). Pour représenter efficacement ce type de données, nous utilisons un *réseau social attribué* (RSA). Le concept de RSA a été introduit et utilisé dans [YGWH10, GTM11]. Il a pour but de représenter les données relationnelles dans lesquelles il y a à la fois les informations de type de contenus (attributs sur les individus) et celles de type de relations (les relations ou interactions entre les individus).

3.3.1 Réseau social attribué

Un réseau social attribué est un réseau social G_s=(V_s, E_s) (V_s est l'ensemble de nœuds et E_s est l'ensemble d'arêtes) augmenté avec un graphe bipartite $G_a = (V_s \cup V_a, E_a)$ qui relie les nœuds dans V_s avec les nœuds dans V_a. Les nœuds dans V_s représentent les acteurs sociaux et s'appellent les *nœuds sociaux*. Les arêtes dans E_s représentent les relations ou interactions entre les acteurs sociaux et s'appellent les *liens sociaux*. Dans ce travail, nous considérons les graphes sociaux non-dirigés (les liens sociaux sont symétriques). Les nœuds dans V_a représentent des attributs sur les acteurs sociaux et s'appellent les *nœuds d'attribut*. Les arêtes E_a dans le graphe bipartite représentent les valeurs connues des attributs sur les acteurs sociaux et s'appellent les *liens d'attribut*. Le concept de RSA est illustré dans la figure 3.1.

Cette représentation est très générique et peut être utilisée de différentes manières selon les données d'entrée, avec ou sans pondération sur les liens. Pour des données tabulaires, les nœuds d'attribut peuvent représenter un attribut et la modalité pourrait être explicitée par un type de lien. Dans notre cas, nous avons fait le choix de représenter chaque modalité de chaque attribut par un nœud d'attribut différent. Nous verrons par la suite (cf. sections suivantes) que pour nos données clientèle, les nœuds d'attribut représenteront alors un type de contrat, féminin ou masculin pour le sexe, etc. Pour les données issues des médias sociaux, nous utiliserons les nœuds d'attribut pour modéliser chacun des mots utilisés dans

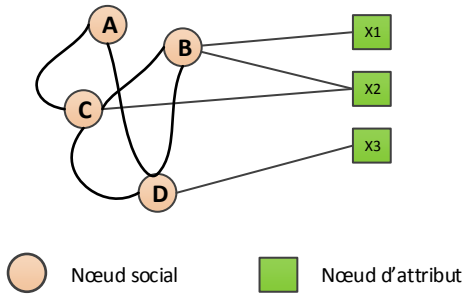


FIGURE 3.1 – Un exemple de réseau social attribué non pondéré

.

les messages, mais nous pourrions également utiliser ces nœuds pour signifier une communauté, un fil de discussion (les liens d'attribut modéliseraient alors la participation aux communautés ou aux fils de discussion).

Concernant la pondération des liens d'attribut et des liens sociaux, cette représentation peut s'utiliser de différentes manières. Par exemple, si le lien social entre (A,B) représente les interactions sociales (messages échangés) entre ces deux individus, le poids du lien peut être le nombre d'interactions (nombre de messages échangés). Si le nœud d'attribut $x1$ représente un contenu publié par l'utilisateur (e.g $x1$ est le mot "Sosh"), le lien entre l'individu A et $x1$ peut être pondéré par le nombre de fois que A a publié $x1$ (e.g le nombre de fois que A a écrit le mot « Sosh »).

3.3.2 Réseau social attribué dans le contexte d'apprentissage incrémental

Nous considérons le cas où les données sont horodatées. Nous avons besoin de construire des modèles prédictifs périodiquement, c'est à dire à des instants prédéfinis. Pour plus de commodité, les instants sont notés par les nombres entier $0, 1, 2, ..., t, ...$; nous appelons aussi chacun de ces instants un *pas de temps* (*time step*). En réalité, ces instants correspondent aux moments où on veut faire des prédictions, par exemple les fins de mois ou de semaines. À l'instant t, nous construisons un RSA désigné par $\mathcal{G}(t)$ à partir des données courantes à cet instant. Pour les média sociaux, les données au pas de temps t concernent tous les interactions sociales ou les contenus créés dans la période entre deux instants $t-1$ et t. De manière intuitive, le RSA $\mathcal{G}(t)$ collecte les événements qui ont eu lieu entre $t-1$ et t : utilisation de mots, interaction avec tel individu, etc. Pour les données tabulaires des autres canaux, les données au pas de temps t sont les valeurs des variables explicatives calculées à l'instant t. La création du RSA $\mathcal{G}$ à partir de ces données est décrite dans la section suivante. Nous avons donc une séquence de graphes attribués $\mathcal{G}(0), \mathcal{G}(1), \mathcal{G}(2)$,..., et on veut effectuer

un apprentissage incrémental à chaque pas de temps $t = 0, 1, 2, ...$

Nous visons à concevoir un algorithme d'apprentissage incrémental pour traiter ce type des données horodatées. Par le terme « apprentissage incrémental », nous désignons les algorithmes qui mettent à jour un modèle à chaque pas de temps t en utilisant seulement les données courantes $\mathcal{G}(t)$. Ce mode d'apprentissage est le contraire de l'apprentissage en mode *batch* qui utilise l'agrégation des toutes les données du passé. L'avantage principal de l'apprentissage incrémental par rapport à l'apprentissage en mode batch est le gain en termes de coût de calcul : à chaque pas de temps, un algorithme incrémental a seulement à traiter les données courantes $\mathcal{G}(t)$ alors qu'un algorithme de batch doit considérer toutes les données dans le passé.

La figure 3.2 présente le concept du RSA dans le contexte d'apprentissage incrémental. Les données à chaque pas de temps t nous permettent de construire un RSA $\mathcal{G}(t)$. Dans ce RSA, les liens sociaux sont des interactions sociales et liens d'attribut représentent les observations sur les individus, ces éléments sont calculés à partir des données courantes au pas de temps t. Nous remarquons que par rapport au RSAs dans les pas de temps précédents, dans $\mathcal{G}(t)$, il y a des nouveaux liens (liens d'attribut ou liens sociaux). Un lien existant à $t-1$ (un lien social ou d'attribut dans $\mathcal{G}(t-1)$) peut disparaître dans le pas de temps t. Nous remarquons aussi qu'il y a des nouveaux nœuds (nœuds sociaux ou nœuds d'attribut). Les nouveaux nœuds sociaux représentent les individus qui viennent d'apparaître et des nouveaux nœuds d'attribut correspondent aux nouveaux attributs que nous observons au pas de temps t.

3.3.3 Représentation des données intercanales avec les RSAs

3.3.3.1 Les données issues des média sociaux

La représentation de données par les RSAs est motivée d'abord par les caractéristiques des données issues des média sociaux : il y a des interactions sociales, des contenus créés par les utilisateurs et ces éléments sont horodatés. Tout d'abord, le graphe est une représentation naturelle des interactions sociales dans les données. Dans les média sociaux, il y a plusieurs types d'information permettant de construire des graphes entre les individus. Les graphes peuvent être construits de façon directe comme un graphe d'amitié (relation d'amitié comme dans Facebook ou *follower-followee* dans Twitter), un graphe de communication (basé sur des échanges des messages entre les utilisateurs - qui communiquent avec qui). Ces graphes peuvent aussi être construits de façon indirecte, par exemple le graphe de *co-liking* (on met une arête entre deux individus s'ils ont cliqué « like » sur un élément commun).

Dans les média sociaux, les individus sont aussi des acteurs qui créent des *contenus*. Ces contenus sont par exemple des éléments du profil déclaré par des utilisateurs, les centres

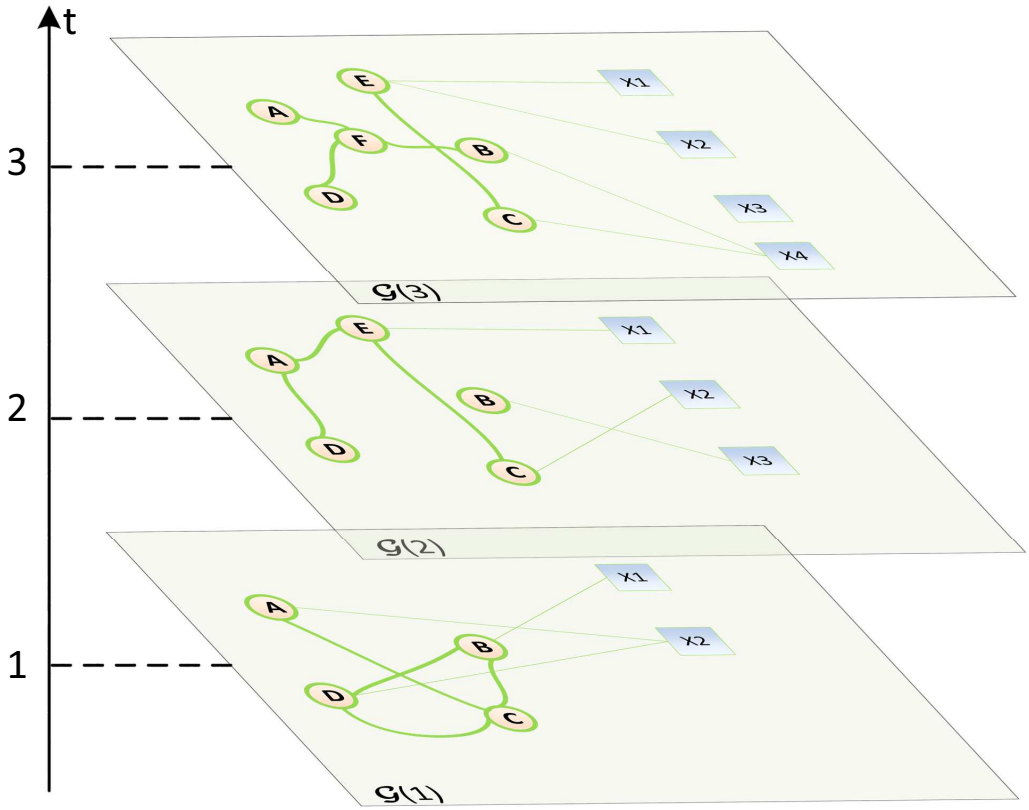


FIGURE 3.2 – Le RSA $\mathcal{G}(t)$ représente les données au pas de temps t.

d'intérêts, les communautés auxquelles ils participent ou encore les messages postés. Pour représenter ces contenus, une représentation tabulaire n'est pas appropriée : les tableaux de données seront en grand nombre de dimensions et très peu denses (beaucoup de valeurs manquantes). Un graphe bipartite acteur-attribut est plus adapté. Les contenus sont transformés en plusieurs attributs, le fait qu'un utilisateur crée les contenus génère les liens vers les attributs. Par exemple, pour représenter les centres d'intérêts des utilisateurs, il suffit de relier les utilisateurs et les intérêts (considérés comme les nœuds d'attribut). Pour le texte, les occurrences des mots dans les textes créés par un utilisateur peuvent être représentées par les liens de cet utilisateur vers des mots (chaque mot correspond à un nœud d'attribut).

Les interactions et les contenus générés par les média sociaux sont datées. Puisque nous nous intéressons à l'apprentissage incrémental, notre approche doit permettre d'intégrer des nouvelles données pour mettre à jour un modèle au moment de l'apprentissage. Pour ce faire, les données issues des média sociaux sont réparties sur la base des intervalles temporels. En réalité, les points de répartition en intervalles correspondent aux moments où l'on

veut effectuer un apprentissage sur les données disponibles et déployer les modèles prédictifs pour faire des prédictions. Nous voyons que ces points de répartitions sont les instants où nous construisons les RSAs.

La figure 3.3 illustre comment les RSAs sont construits à partir des média sociaux. Nous avons 3 nœuds sociaux (3 individus dans les média sociaux). Nous disposons des données dans 3 périodes, séparées par les instants $t = 0, 1, 2, 3$. Nous utilisons les termes période 1, période 2, période 3 pour désigner ces trois périodes. Les données contiennent les interactions sociales et les contenus générés par chaque individu dans chaque période (à gauche de la figure). Il y a 3 types de contenus désignés par X, Y et Z. Dans les cas où les contenus créés par les utilisateurs sont des textes, X, Y, Z correspondent aux mots utilisés par les utilisateurs dans leurs textes. Dans la période t, nous créons les liens sociaux correspondant aux interactions sociales entre les individus au cours de cette période (e.g $A - B$ dans la période 1, $A - C$ dans la période 2, $A - B, B - C$ dans la période 3) ; nous créons aussi les liens d'attribut correspondant aux contenus publiés par chaque individu. Par exemple, les liens $A - X, A - Y, B - X$ dans la période 1 indiquent que A a écrit les mots X et Y dans ses textes et B a écrit le mot X dans cette période. À chaque nouveau pas de temps, nous avons aussi de nouveaux nœuds sociaux (par exemple C apparaît au cours de la période 2) et de nouveaux nœuds d'attribut (par exemple Z est utilisé pour la première fois dans la période 2, respectivement T dans la période 3). Remarquons aussi qu'un nœud d'attribut existant n'est pas forcément connecté à un nœud social. Par exemple, l'attribut X n'est pas utilisé dans la période 2. On garde X dans le graphe parce qu'il est utilisé dans la période 3.

3.3.3.2 Les données attribut-valeur

Les données tabulaires issues des autres canaux (dans le contexte du CRM intercanal) peuvent être intégrées dans les graphes bipartites individu-attribut. Par exemple, à partir de la base de données clientèle de l'entreprise, nous pouvons extraire des variables concernant la consommation (calculée sur une fenêtre temporelle avant t), la durée restante de contrat, le type de service souscrit par les clients, etc. Les variables peuvent être calculées à chaque pas de temps t pour chaque client.

Les variables dans les données tabulaires sont catégorielles ou continues. Nous transformons d'abord les variables continues en variables catégorielles par la discrétisation. Pour intégrer les variables catégorielles dans les RSAs, nous créons un nœud d'attribut pour chacune des modalités des variables.

La figure 3.4 illustre comment les données attribut-valeur sont intégrées dans les RSAs. Nous avons une variable caractérisant la consommation du client dans une fenêtre temporelle (par exemple d'un mois) juste avant l'instant t. Cette variable a deux modalités : consommation élevée ou consommation faible. Dans le RSA, nous créons donc 2 nœuds d'at-

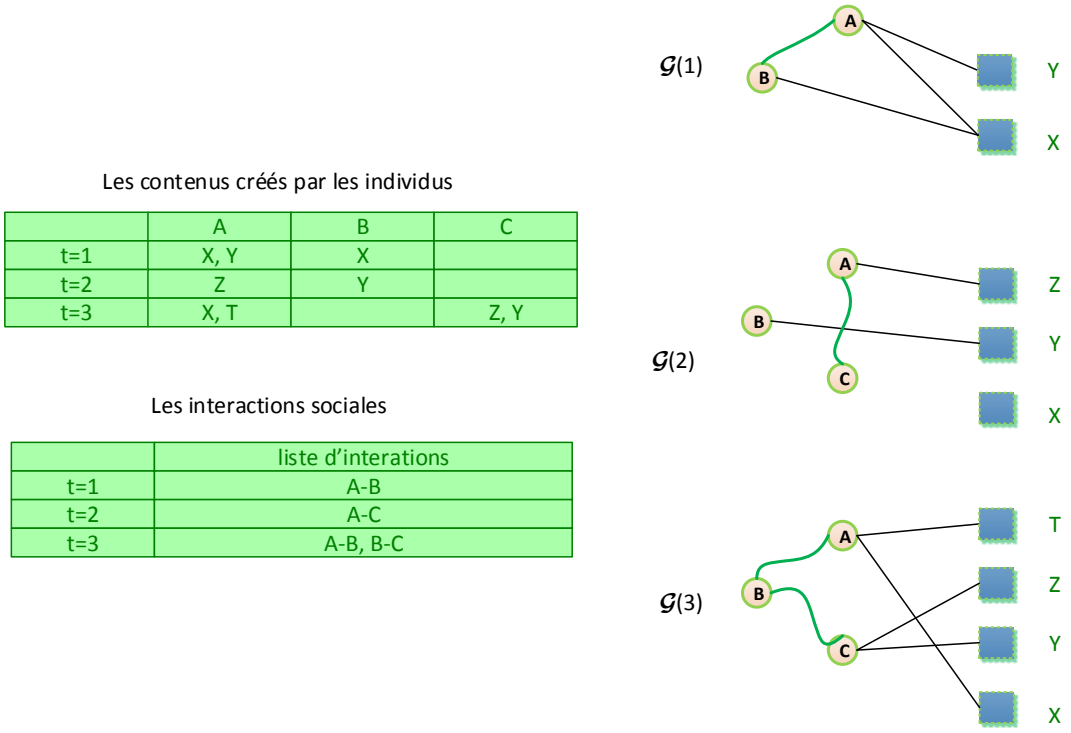


FIGURE 3.3 – Construction des RSAs à partir des données issues des média sociaux.

tribut correspondant à ces 2 modalités *conso_elevee* et *conso_faible*. À chaque pas de temps t, $(t = 1, 2, 3)$, la variable consommation est calculée à partir des données client. Dans le RSA $\mathcal{G}(t)$ $(t = 1, 2, 3)$ nous relions les individus vers les nœuds d'attribut correspondant. Prenons par exemple l'instant $t = 2$, nous voyons qu'il y a les liens d'attribut $A - conso\_faible$, $B - conso\_elevee$ et $C - conso\_elevee$. Ces liens d'attribut indiquent que la valeur de la variable consommation de A est faible, celles de B et C sont élevées à cet instant.

3.4 Notre problème d'apprentissage incrémental

Dans cette section, nous définissons notre problème d'apprentissage incrémental. Nous expliquons aussi le rôle de ce problème dans le contexte d'une stratégie du CRM intercanale et les mesures de l'engagement.

3.4.1 Description du problème

Notre problème est d'apprendre un modèle à facteurs latents (les facteurs latents des individus) à chaque pas de temps t à partir des données sous forme de RSA. Autrement dit, il s'agit de transformer les données sous forme de graphes attribués à une représentation

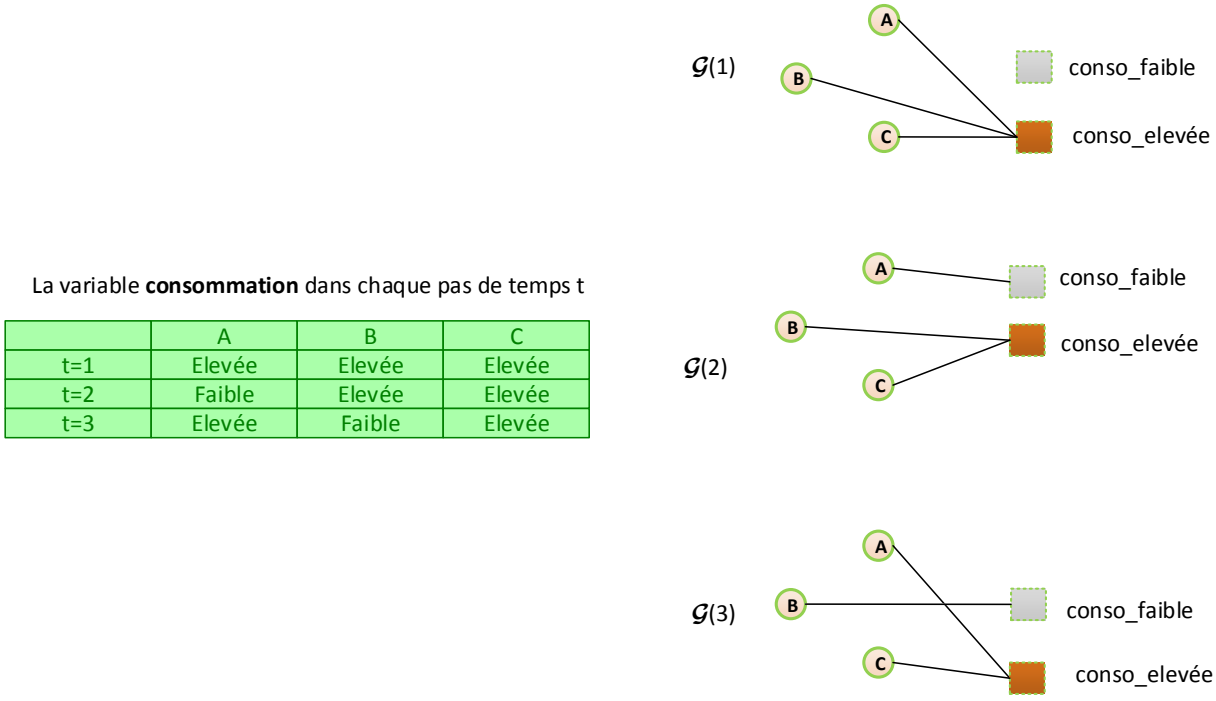


FIGURE 3.4 – Construction des RSAs à partir des données attribut-valeur.

tabulaire (les facteurs latents).

Nous nous intéressons à l'apprentissage incrémental. Nous avons une séquence de RSA $\mathcal{G}(0), \mathcal{G}(1), \mathcal{G}(2)$,..., chaque $\mathcal{G}(t)$ représente les données collectées au pas de temps t. Étant donné que nous avons un modèle $\mathcal{M}(t-1)$ (les facteurs latents) appris à partir des données jusqu'à $t-1$, le problème est de mettre à jour $\mathcal{M}(t-1)$ avec les données à t, c'est-à-dire $\mathcal{G}(t)$, pour obtenir le nouveau modèle $\mathcal{M}(t)$ à t.

Notre problème peut être considéré comme un problème d'*apprentissage de la représentation* dans un contexte incrémental. L'apprentissage de la représentation [BCV12] est un sous-domaine de l'apprentissage automatique qui étudie les techniques qui transforment des données brutes en une représentation qui peut être efficacement exploitée dans une tâche d'apprentissage classique (e.g classification supervisée comme dans [LY09] ou prédiction de liens comme dans [GDG12]). La particularité de notre problème est la nécessité d'effectuer un apprentissage *incrémental* de représentation, problème qui, à notre connaissance, n'a pas encore été considéré dans la littérature de manière dédiée.

3.4.2 Le problème dans le contexte de la thèse

Le défi de la stratégie du CRM intercanal (i.e l'exploitation simultanée des données clients et des données média sociaux) est adressé par la représentation RSA. Comme men-

tionné dans la section 3.3.3, cette représentation peut représenter à la fois les données attribut-valeur et les relations ou interactions. À partir des RSAs construits, nous calculons les facteurs latents à chaque pas de temps. Les facteurs latents appris peuvent être considérés comme des variables explicatives (qui caractérisent les individus dans le RSA) à chaque période. Ces variables explicatives caractérisent les données issues de multiples sources, y compris les média sociaux. Ces variables peuvent être ensuite utilisées pour les tâches d'apprentissage classique : classification, régression, etc. Dans l'objectif de la thèse, nous nous intéressons à prédire des variables (cibles) qui caractérisent l'engagement des clients avec l'entreprise. Ces variables cibles correspondent au comportement *futur* du client (le fait de churner, migrer, émettre d'un Tweet lié à la marque...) Nous proposons d'utiliser les facteurs latents calculés comme variables explicatives à chaque période pour construire les modèles prédictifs.

3.5 Apprentissage incrémental des modèles à facteurs latents pour les réseaux sociaux attribués

3.5.1 Apprentissage des facteurs latents à partir d'un réseau social attribué statique

Dans ce paragraphe, nous décrivons l'apprentissage des facteurs latents à partir d'un RSA en mode hors ligne. Notre méthode est une adaptation des méthodes de FM décrites dans la section 3.1.3 pour un RSA.

Nous considérons un RSA $\mathcal{G} = (G_s, G_a)$, où G_s=(V_s, E_s) est le graphe social qui a $n_s = |V_s|$ nœuds sociaux et G_a=$(V_s \cup V_a, E_a)$ qui a $n_a = |V_a|$ nœuds d'attribut. Dans la suite, nous notons $\mathbf{S}$ la matrice d'adjacence du graphe social et $\mathbf{A}$ est la matrice d'adjacence du graphe bipartite d'attribut. Autrement dit, $\mathbf{S}_{ij}$ est le poids du lien social (i, j) (binaire dans le cas où le graphe social n'est pas pondéré) et $\mathbf{A}_{ik}$ est le poids du lien d'attribut (i, k) (binaire dans le cas des liens d'attribut non pondérés, ceux qui modélisent les liens d'attribut clientèle ; le poids est un entier pour les liens d'attribut sociaux dans le cas nous considérerons des attributs mots).

D'après l'approche à facteurs latents, chacun des nœuds (nœuds sociaux ou nœuds d'attribut) est caractérisé par un vecteur de variables latentes (réelles) de dimension d (un paramètre du modèle). Nous notons u_i le vecteur des facteurs latents de l'i-ième nœud social et $\mathbf{U}$ la matrice constituée des u_i, $(i = 1, 2, ..., n_s)$, p_k le vecteur des facteurs latents du k-ième nœud d'attribut et $\mathbf{P}$ la matrice constituée des p_k $(k = 1, 2, ..., n_a)$ (par commodité, dans ce texte nous appelons l'i-ième nœud social le nœud social i et le k-ième nœud d'attribut le nœud d'attribut k). Les méthodes de FM consistent à décomposer simultanément les ma-

trices d'adjacence des graphes G_s et G_a pour obtenir des matrices de rang d $\mathbf{U}$ et $\mathbf{P}$.

3.5.1.1 Factorisation relationnelle régularisée de matrice (FRRM)

Selon l'approche FRRM [LY09], les matrices $\mathbf{U}$ et $\mathbf{P}$ sont déterminées en minimisant la fonction d'objectif suivante :

$$\begin{aligned} Q_{FRRM}\left(\mathbf{U}, \mathbf{P}|\mathcal{G}\right) = \frac{1}{2}\sum_{(i,k)\in E_a}\left(\mathbf{A}_{ik} - u_i p_k^T\right)^2 + \frac{\alpha}{2}\sum_{(i,j)\in E_s}\mathbf{S}_{ij}\left\|u_i - u_j\right\|^2 \\ + \frac{\lambda}{2}\left(\sum_{i=1}^{n_s}\|u_i\|^2 + \sum_{k=1}^{n_a}\|p_k\|^2\right) \end{aligned} \tag{3.5}$$

Nous remarquons que la FRRM définie dans l'équation 3.5 est aussi une décomposition de la matrice d'adjacence du graphe d'attribut $\mathbf{A}$. La différence entre FRRM et la version originale de la FM (définie dans l'équation 3.3) est dans le deuxième terme - le terme de régularisation relationnelle. Comme mentionné avant, ce terme dans la fonction d'objectif permet de minimiser les distances (dans l'espace latent) entre les individus connectés dans le graphe social. α est un paramètre du modèle permettant de régler la contribution du graphe social dans la factorisation. L'approche FRRM suppose que les liens dans le graphe social possèdent la caractéristique d'*homophilie* (décrite dans la section 2.3.1), c'est-à-dire les acteurs sociaux connectés ont tendance à avoir les profils similaires et donc ils auront une grande probabilité d'avoir les mêmes actions.

Remarque Le terme de régularisation relationnelle peut être réécrite comme suit (on considère $\mathbf{S}_{ij} = 0$ s'il n'y pas de lien entre i et j) :

$$\begin{aligned} \frac{1}{2}\sum_{(i,j)\in E_s}\mathbf{S}_{ij}\left\|u_i - u_j\right\|^2 = \\ &= \frac{1}{2}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s}\mathbf{S}_{ij}\left(u_i u_i^T + u_j u_j^T - 2u_j u_j^T\right) \\ &= \frac{1}{2}\left(\sum_{i=1}^{n_s}\mathbf{D}_{ii}u_i u_i^T + \sum_{j=1}^{n_s}\mathbf{D}_{jj}u_j u_j^T - 2\sum_{i=1}^{n_s}\sum_{j=1}^{n_s}\mathbf{S}_{ij}u_i u_j^T\right) \\ &= \frac{1}{2}\left(\mathbf{Tr}(\mathbf{U}^T\mathbf{D}\mathbf{U}) + \mathbf{Tr}(\mathbf{U}^T\mathbf{D}\mathbf{U}) - 2\mathbf{Tr}(\mathbf{U}^T\mathbf{S}\mathbf{U})\right) \\ &= \mathbf{Tr}(\mathbf{U}^T\mathbf{L}\mathbf{U}) \end{aligned} \tag{3.6}$$

où $\mathbf{L} = \mathbf{D} - \mathbf{S}$ est la matrice Laplacienne [Chu99] du graphe social, $\mathbf{D}$ est la matrice

diagonale dont les éléments diagonaux sont les dégrés des nœuds dans le graphe social $\mathbf{D}_{jj} = \sum_{i=1}^{n_s} \mathbf{S}_{ij}$.

Comme constaté dans [LY09], il est souvent préférable d'utiliser la matrice Laplacienne normalisée à la place de la matrice Laplacienne. La matrice Laplacienne normalisée est définie comme suit :

$$\mathbf{L}_{norm} = \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}} \tag{3.7}$$

Le terme de régularisation relationnelle devient donc

$$\mathbf{Tr}(\mathbf{U}^T \mathbf{L}_{norm} \mathbf{U}) = \frac{1}{2} \sum_{(i,j) \in E_s} S_{ij} \left\| \frac{u_i}{\sqrt{\mathbf{D}_{ii}}} - \frac{u_j}{\sqrt{\mathbf{D}_{jj}}} \right\|^2 \tag{3.8}$$

Ce terme de régularisation a le même but que celui du terme de régularisation non-normalisé (défini dans l'équation 3.4) : exploiter les relations sociales de type *homophilie*, mais ici les vecteurs latents des nœuds sociaux sont normalisés par les racines carrées des dégrés de ces nœuds dans le graphe social. Comme dans le travail de [LY09], dans nos expérimentations, nous avons essayé le FRRM avec ce terme de régularisation. Nous avons observé que, ce terme de régularisation normalisé donne souvent une meilleure performance que le terme non-normalisé.

3.5.1.2 Factorisation collective de matrices (FCM)

Une autre approche de FM pour exploiter les deux matrices **S** et **A** est la FCM [SG08]. Elle consiste à décomposer simultanément ces deux matrices en partageant les facteurs sur les deux décompositions. Une façon de définir cette factorisation est comme suit (nous utilisons les mêmes notations comme dans l'équation 3.5) :

$$\begin{aligned} Q_{FCM}(\mathbf{U}, \mathbf{P} | \mathcal{G}) = & \frac{1}{2} \sum_{(i,k) \in E_a} \left(\mathbf{A}_{ik} - u_i p_k^T \right)^2 + \frac{\alpha}{2} \sum_{(i,j) \in E_s} \left(\mathbf{S}_{ij} - u_i^T u_j \right)^2 \\ & + \frac{\lambda}{2} \left(\sum_{i=1}^{n_s} \|u_i\|^2 + \sum_{k=1}^{n_a} \|p_k\|^2 \right) \end{aligned} \tag{3.9}$$

Le premier terme correspond à la décomposition de la matrice **A**, le deuxième terme correspond à la décomposition de la matrice **S** et le troisième terme est un terme de régularisation. Il n'y a pas d'interprétation intuitive pour la FCM (comme avec la FRRM), mais il a été démontré qu'elle donne de bonnes performances de prédiction dans certaines applications (par exemple la classification des pages web en utilisant les hyperliens et les contenus

des pages [ZYCG07]). Selon l'analyse empirique de [Li10, chapitre 3], la décomposition simultanée de la matrice de contenu et de la matrice du graphe social est adaptée aux liens sociaux qui possèdent la caractéristique d'*équivalence stochastique* (cf. la section 2.3.1).

Dans les expérimentations menées dans cette thèse, nous sommes basés principalement sur l'hypothèse d'homophile des graphes sociaux. Les graphes sont aussi construits sous l'hypothèse d'homophilie (le cas de données synthétiques). Par conséquence, nous nous intéressons d'abord à tester l'approche FRRM.

3.5.2 Apprentissage incrémental des facteurs latents

Dans notre problème d'apprentissage, nous avons besoin d'apprendre les facteurs latents des nœuds à chaque pas de temps t. On pourrait utiliser un apprentissage qui utilise l'agrégation des données jusqu'à t :

$$\mathbf{U}^{\star}(t), \mathbf{P}^{\star}(t) = \arg\min_{\mathbf{U},\mathbf{P}} Q(\mathbf{U}, \mathbf{P} | \mathcal{G}^{agg}(t)) \tag{3.10}$$

où $\mathcal{G}^{agg}(t)$ désigne le RSA construits à partir de toutes les données dans le passé jusqu'à t et Q est une fonction objectif pour la factorisation des matrices (définie dans les paragraphes précédents).

Mais ici, nous nous intéressons à l'apprentissage incrémental selon lequel les facteurs latents sont appris à partir des données courantes à chaque pas de temps t (i.e $\mathcal{G}(t)$) en réutilisant les facteurs latents appris à la période précédente. Nous proposons d'apprendre les facteurs latents à t par la fonction objectif suivante :

$$\begin{aligned} Q_{inc}(\mathbf{U}, \mathbf{P}, t) = & \\ & Q(\mathbf{U}, \mathbf{P} | \mathcal{G}(t)) \\ & + \frac{\mu}{2} \left(\sum_{i \in V_s(t-1)} \|u_i - u_i^{\star}(t-1)\|^2 + \sum_{k \in V_a(t-1)} \|p_k - p_k^{\star}(t-1)\|^2 \right) \end{aligned} \tag{3.11}$$

où $V_s(t-1)$ et $V_a(t-1)$ sont respectivement l'ensemble des nœuds sociaux et l'ensemble des nœuds d'attribut dans la période précédente ; $u_i^{\star}(t-1)$ et $p_k^{\star}(t-1)$ sont les vecteurs latents du nœud social i et du nœud d'attribut k appris au pas de temps précédent ; μ est un paramètre de notre méthode incrémentale.

Cette fonction objectif contient deux termes. Le premier terme correspond à un MFL avec le RSA incrémental $\mathcal{G}(t)$. Le second terme est un terme de régularisation. Ce terme réutilise les facteurs latents appris au pas de temps précédent. Intuitivement, il a pour but de réduire au minimum les déplacements des nœuds existants dans l'espace latent en passant à un

nouveau pas de temps. Avec ces deux termes dans la fonction objectif, nous effectuons un apprentissage des facteurs latents à partir des données courantes (représentées par $\mathcal{G}(t)$) et du modèle dans le passé (les facteurs latents appris à $t-1$). Les facteurs latents des nœuds existants sont mis à jour dans le modèle actuel. Le paramètre μ permet de régler la contribution du modèle du passé dans le modèle actuel.

Le terme de régularisation permet aussi de conserver l'espace latent dans le passage d'un pas de temps au pas de temps suivant. Cela assure que les sémantiques des variables latentes restent les mêmes au fil du temps. Cela est un avantage de notre méthode. Les facteurs latents appris à chaque pas de temps peuvent être utilisées comme les variables explicatives pour prédire des variables cibles. On peut par exemple apprendre un classifier avec les facteurs latents à $t-1$ et ensuite le déployer avec les facteurs latents à t.

3.6 Algorithme d'optimisation et sa complexité

3.6.1 Algorithme d'optimisation

Nous avons formalisé le problème d'apprentissage des facteurs latents comme un problème d'optimisation. L'apprentissage en mode hors ligne consiste à minimiser la fonction objectif définie dans l'équation 3.10 et l'apprentissage incrémental correspond à celle définie dans l'équation 3.11. Nous utilisons un algorithme d'optimisation très connu : l'algorithme des *moindres carrés en alternance* (MCA) (*alternating least square*). Le MCA est un algorithme parallélisable et a été utilisé pour les problèmes de filtrage collaboratif à grande échelle [ZWSP08].

Pour optimiser une fonction objectif de plusieurs variables, le MCA met à jour alternativement une variable (ou un ensemble de variables) tout en fixant les autres. Dans notre cas, les variables sont les facteurs latents des nœuds (les u_i et les p_k).

L'algorithme 3.1 est la version adaptée de l'algorithme MCA pour notre problème d'apprentissage.

L'idée derrière cet algorithme est de mettre à jour de manière itérative les vecteurs latents des nœuds, un par un jusqu'à convergence. Pour mettre à jour le vecteur latent d'un nœud (nœud social ou nœud d'attribut), l'algorithme fixe les facteurs latents de tous les autres nœuds et minimise la fonction objectif en fonction des facteurs latents de ce nœud. Ceci est possible parce que la fonction objectif Q est convexe lorsque l'on considère uniquement les facteurs latents d'un nœud. Cette convexité partielle est facile à prouver : Q est la somme des termes au carré, chacun de ces termes est convexe en ce qui concerne les facteurs latent d'un nœud - u_i ou p_k. Par exemple, nous montrons dans la suite que la fonction objectif de RRMF (l'équation 3.5) est convexe en ce qui concerne u_i. En enlevant tous les termes qui ne

Algorithm 3.1 L'algorithme MCA pour l'apprentissage des facteurs latents

Entrée : une fonction objectif $Q(\mathbf{U}, \mathbf{P})$, $\mathbf{U}$ et $\mathbf{P}$ sont des matrices composés des facteurs latents de taille $n_s \times d$ et $n_a \times d$, respectivement
Résultat : un optimum local de Q

```
Initialiser les facteurs latents (i.e U, P)(de manière aléatoire)
répéter
    pour tous i ∈ {1, 2, ..., n_s} faire
        u_i ← arg min_{u_i} Q(U, P)        # u_i est la ligne i de U
    fin pour
    pour tous k ∈ {1, 2, ..., n_a} faire
        p_k ← arg min_{p_k} Q(U, P)        # p_k est la ligne k de P
    fin pour
jusqu'à ce que le critère de convergence soit satisfait
```

concernent pas u_i, nous devons montrer que la fonction suivante est convexe :

$$Q_{FRRM}(u_i) = \frac{\alpha}{2} \sum_{j \in \mathcal{N}_s(i)} \mathbf{S}_{ij} \left(u_i - u_j\right)^2 + \frac{1}{2} \sum_{k \in \mathcal{N}_a(i)} \left(A_{ik} - u_i p_k^T\right)^2 + \frac{\lambda}{2} \|u_i\|^2$$

où $\mathcal{N}_s(i)$ est l'ensemble de tous les voisins sociaux du nœud i, (nœuds sociaux connectés au nœud i) et $\mathcal{N}_a(i)$ est l'ensemble de tous les voisins d'attribut du nœud i, (nœuds d'attribut connectés au nœud i) dans le RSA $\mathcal{G}$. Nous réécrivons cette fonction comme ci-dessous :

$$Q_{FRRM}(u_i) = \frac{\alpha}{2} \sum_{j \in \mathcal{N}_s(i)} \mathbf{S}_{ij} \left\|u_i - u_j\right\|^2 + \frac{1}{2} \sum_{k \in \mathcal{N}_a(i)} \left(A_{ik}^2 - 2A_{ik} u_i p_k^T + u_i p_k^T p_k u_i^T\right) + \frac{\lambda}{2} \|u_i\|^2$$

La matrice hessienne de $Q_{FRRM}(u_i)$ est

$$H(Q_{FRRM}(u_i)) = \alpha \sum_{j \in \mathcal{N}_s(i)} \mathbf{S}_{ij} \mathbf{I}_d + \sum_{k \in \mathcal{N}_a(i)} p_k^T p_k + \lambda \mathbf{I}_d$$

$$\Rightarrow H(Q_{FRRM}(u_i)) = \sum_{k \in \mathcal{N}_a(i)} p_k^T p_k + \left(\lambda + \alpha \mathbf{D}_{ii}\right) \mathbf{I}_d$$

où $\mathbf{D}_{ii}$ est le dégré du nœud i dans le graphe social ($\mathbf{D}_{ii} = \sum_{j=1}^{n_s} \mathbf{S}_{ij} = \sum_{j \in \mathcal{N}_s(i)} \mathbf{S}_{ij}$) et $\mathbf{I}_d$ est la matrice d'identité de taille $d \times d$. Cette matrice est définie positive car :

$$\forall \mathbf{x} \in \mathbb{R}^d : \mathbf{x} H(Q_{FRRM}(u_i)) \mathbf{x}^T = \sum_{k \in \mathcal{N}_a(i)} \left(\mathbf{x} p_k^T\right)^2 + \left(\lambda + \alpha \mathbf{D}_{ii}\right) \|\mathbf{x}\|^2 \geq 0$$

d'où nous concluons que la fonction $Q_{FRRM}(u_i)$ est convexe.

D'une manière similaire, nous pouvons prouver la convexité partielle de la fonction objectif pour les autres cas (pour p_k, pour CMF et pour les fonctions objectif d'apprentissage incrémental).

La convergence de l'algorithme MCA peut être prouvée facilement : à chaque mise à jour, Q diminue et Q est inférieurement bornée par 0. Puisque la fonction Q n'est pas convexe en général (en ce qui concerne toutes les variables), l'algorithme converge vers un optimum local. En pratique, lors de nos expérimentations, nous avons observé qu'un optimum local est atteint au bout d'une vingtaine d'itérations : la performance de prédiction devient stable. Nous utilisons donc le nombre d'itérations comme critère de convergence (valeur de 20). Notons que dans un autre contexte applicatif, avec d'autres données, le nombre d'itérations assurant la convergence devra être redéfini.

3.6.2 Les règles de mise à jour les facteurs latents

Nous présentons les règles de mise à jour des facteurs latents de chacun des nœuds dans chaque interaction de de l'algorithme 3.1. Ces mises à jour correspondent aux lignes 4 et 7 de cet algorithme.

3.6.2.1 FRRM

Nous considérons premièrement la fonction objectif de FRRM pour l'apprentissage hors ligne définie dans l'équation 3.5.

Mise à jour des facteurs latents d'un nœud social (u_i) Nous devons calculer l'optimum de Q_{FRRM} en ce qui concerne u_i. En enlevant tous les termes qui ne concernent pas u_i, le problème devient la minimisation de la fonction suivante :

$$Q_{FRRM}(u_i) = \frac{\alpha}{2} \sum_{j \in \mathcal{N}_s(i)} \mathbf{S}_{ij} \left\| u_i - u_j \right\|^2 + \frac{1}{2} \sum_{k \in \mathcal{N}_a(i)} \left(A_{ik} - u_i p_k^T \right)^2 + \frac{\lambda}{2} \left\| u_i \right\|^2$$

où $\mathcal{N}_s(i)$ est l'ensemble de tous les voisins sociaux du nœud i, (nœuds sociaux connectés au nœud i) et $\mathcal{N}_a(i)$ est l'ensemble de tous les voisins d'attribut du nœud i, (nœuds d'attribut connectés au nœud i) dans le RSA $\mathcal{G}$. Comme $Q^{(i)}_{FRRM}$ est une fonction convexe de u_i, elle a toujours un optimum. Cet optimum est le point où le gradient est null :

$$\frac{\partial Q_{FRRM}(u_i)}{\partial \mathbf{U}_{im}} = 0, \qquad m = 1, 2, .., d$$

(Nous rappellons que u_i est la ligne i de $\mathbf{U}$ et p_k est la ligne k de $\mathbf{P}$)

$$\Rightarrow \alpha \sum_{j \in \mathcal{N}_s(i)} S_{ij} \left(\mathbf{U}_{im} - \mathbf{U}_{jm} \right) - \sum_{k \in \mathcal{N}_a(i)} \left(\mathbf{A}_{ik} - u_i p_k^T \right) \mathbf{P}_{km} + \lambda \mathbf{U}_{im} = 0, \quad m = 1, 2, ...d$$

$$\Rightarrow \alpha \sum_{j \in \mathcal{N}_s(i)} \mathbf{S}_{ij} \left(u_i - u_j \right) - \sum_{k \in \mathcal{N}_a(i)} \left(\mathbf{A}_{ik} p_k - u_i p_k^T p_k \right) + \lambda u_i = 0$$

$$\Rightarrow u_i \left(\sum_{k \in \mathcal{N}_a(i)} p_k^T p_k + \left(\lambda + \alpha \mathbf{D}_{ii} \right) \mathbf{I}_d \right) = \left(\alpha \sum_{j \in \mathcal{N}_s(i)} \mathbf{S}_{ij} u_j + \sum_{k \in \mathcal{N}_a(i)} \mathbf{A}_{ik} p_k \right)$$

où $\mathbf{D}_{ii}$ est le dégrée du nœud i dans le graphe social ($\mathbf{D}_{ii} = \sum_{j=1}^{n_s} \mathbf{S}_{ij} = \sum_{j \in \mathcal{N}_s(i)} \mathbf{S}_{ij}$) et $\mathbf{I}_d$ est la matrice d'identité de taille $d \times d$. Le règle de mise à jour pour u_i est :

$$u_i \leftarrow \left(\alpha \sum_{j \in \mathcal{N}_s(i)} \mathbf{S}_{ij} u_j + \sum_{k \in \mathcal{N}_a(i)} \mathbf{A}_{ik} p_k \right) \left(\sum_{k \in \mathcal{N}_a(i)} p_k^T p_k + \left(\lambda + \alpha \mathbf{D}_{ii} \right) \mathbf{I}_d \right)^{-1} \tag{3.12}$$

Mise à jour des facteurs latents d'un nœud d'attribut (p_k) Pour mettre à jour les facteurs latents d'un nœud d'attribut k, nous cherchons l'optimum de Q en fonction de p_k. Ce la est équivalent à minimiser :

$$Q_{FRRM}(p_k) = \sum_{i \in \mathcal{N}_s(k)} \left(\mathbf{A}_{ik} - p_k u_i^T \right)^2 + \lambda \left\| p_k \right\|^2$$

où $\mathcal{N}_s(k)$ est l'ensemble de nœuds sociaux connectés au nœud d'attribut k. En cherchant le point où le gradient est nulle, nous trouvons le règle de mise à jour suivant :

$$p_k \leftarrow \left(\sum_{i \in \mathcal{N}_s(k)} \mathbf{A}_{ik} u_i \right) \left(\sum_{i \in \mathcal{N}_s(k)} u_i^T u_i + \lambda \mathbf{I}_d \right)^{-1} \tag{3.13}$$

3.6.2.2 FCM

Nous considérons la fonction objectif définie dans l'équation 3.9. En utilisant les mêmes manipulations que dans les paragraphes précédents, nous avons les règles de mise à jour suivants.

Mise à jour des facteurs latents d'un nœud social (u_i)

$$u_i \leftarrow \left(\alpha \sum_{j \in \mathcal{N}_s(i)} \mathbf{S}_{ij} u_j + \sum_{k \in \mathcal{N}_a(i)} \mathbf{A}_{ik} p_k \right) \left(\alpha \sum_{j \in \mathcal{N}_s(i)} u_j^T u_j + \sum_{k \in \mathcal{N}_a(i)} p_k^T p_k + \lambda \mathbf{I}_d \right)^{-1} \tag{3.14}$$

Mise à jour des facteurs latents d'un nœud d'attribut (p_k) Le même règle que dans le cas de FRRM (cf. le règle 3.13).

3.6.2.3 Apprentissage incrémental avec l'approche FRRM

Nous considérons la fonction objectif dans l'équation 3.11 dans le cas Q est remplacé par Q_{FRRM}. Avec la présence du terme de régularisation

$$\frac{\mu}{2} \left(\sum_{i \in V_s(t-1)} \| u_i - u_i^\star (t-1) \|^2 + \sum_{k \in V_a(t-1)} \| p_k - p_k^\star (t-1) \|^2 \right)$$

les règles de mise à jour deviennent les suivants (nous remarquons que les voisinages $\mathcal{N}_s$, $\mathcal{N}_a$ et les dégrés $\mathbf{D}_{ii}$ sont définis sur le RSA $\mathcal{G}(t)$) :

Mise à jour des facteurs latents d'un nœud social (u_i)

– Si $i \in V_s(t-1) \cap V_s(t)$

$$u_i \leftarrow \left(\alpha \sum_{j \in \mathcal{N}_s(i)} \mathbf{S}_{ij} u_j + \sum_{k \in \mathcal{N}_a(i)} \mathbf{A}_{ik} p_k + \mu u_i^\star (t-1) \right) \left(\sum_{k \in \mathcal{N}_a(i)} p_k^T p_k + (\lambda + \alpha \mathbf{D}_{ii} + \mu) \mathbf{I}_d \right)^{-1} \tag{3.15}$$

– Sinon

$$u_i \leftarrow \left(\alpha \sum_{j \in \mathcal{N}_s(i)} \mathbf{S}_{ij} u_j + \sum_{k \in \mathcal{N}_a(i)} \mathbf{A}_{ik} p_k \right) \left(\sum_{k \in \mathcal{N}_a(i)} p_k^T p_k + (\lambda + \alpha \mathbf{D}_{ii}) \mathbf{I}_d \right)^{-1} \tag{3.16}$$

Mise à jour des facteurs latents d'un d'attribut (p_k)

– Si $k \in V_s(t-1) \cap V_a(t)$

$$p_k \leftarrow \left(\sum_{i \in \mathcal{N}_s(k)} \mathbf{A}_{ik} u_i + \mu p_k^\star (t-1) \right) \left(\sum_{k \in \mathcal{N}_s(i)} u_i^T u_i + (\lambda + \mu) \mathbf{I}_d \right)^{-1} \tag{3.17}$$

– Sinon

$$p_k \leftarrow \left(\sum_{i \in \mathcal{N}_s(k)} \mathbf{A}_{ik} u_i \right) \left(\sum_{k \in \mathcal{N}_s(i)} u_i^T u_i + \lambda \mathbf{I}_d \right)^{-1} \tag{3.18}$$

3.6.2.4 Apprentissage incrémental avec l'approche FCM

Nous considérons la fonction objectif dans l'équation 3.11 dans le cas Q est remplacé par Q_{FCM}. Les règles de mise à jour deviennent les suivants :

Mise à jour des facteurs latents d'un nœud social (u_i)

– Si $i \in V_s(t-1) \cap V_s(t)$

$$\begin{aligned} u_i \leftarrow & \left(\alpha \sum_{j \in \mathcal{N}_s(i)} \mathbf{S}_{ij} u_j + \sum_{k \in \mathcal{N}_a(i)} \mathbf{A}_{ik} p_k + \mu u_i^\star(t-1) \right) \\ & \left(\alpha \sum_{j \in \mathcal{N}_s(i)} u_j^T u_j + \sum_{k \in \mathcal{N}_a(i)} p_k^T p_k + (\lambda + \mu) \mathbf{I}_d \right)^{-1} \end{aligned} \tag{3.19}$$

– Sinon

$$u_i \leftarrow \left(\alpha \sum_{j \in \mathcal{N}_s(i)} \mathbf{S}_{ij} u_j + \sum_{k \in \mathcal{N}_a(i)} \mathbf{A}_{ik} p_k \right) \left(\alpha \sum_{j \in \mathcal{N}_s(i)} u_j^T u_j + \sum_{k \in \mathcal{N}_a(i)} p_k^T p_k + \lambda \mathbf{I}_d \right)^{-1} \tag{3.20}$$

Mise à jour des facteurs latents d'un d'attribut (p_k) Le même règle que celui dans le cas de l'apprentissage incrémental avec FRRM (cf 3.17 et 3.18) .

3.6.3 Analyse de complexité

Nous estimons le nombre d'opérations multiplicatives (de deux nombres réels) à chaque itération de l'algorithme 3.1. Chaque itération se compose de la mise à jour des facteurs latents de tous les nœuds dans le RSA, avec des règles de mise à jour pour chacun des nœuds définies dans la section précédente 3.6.2.

Nous remarquons que toutes les règles de mise à jour (pour FRRM ou FCM, apprentissage hors-ligne ou incrémental) sont de même forme : $b \leftarrow a \times \mathbf{C}^{-1}$, où a, b sont des vecteurs de taille d (le nombre de dimensions latentes). La matrice $\mathbf{C}$ varie selon la méthode (voir la section 3.6.2). Par exemple, pour mettre à jour les facteurs latents u_i dans l'approche FRRM,

nous avons :

$$\mathbf{C} = \sum_{k \in \mathcal{N}_a(i)} p_k^T p_k + (\lambda + \alpha \mathbf{D}_{ii})\, \mathbf{I}_d$$
$$a = \alpha \sum_{j \in \mathcal{N}_s(i)} \mathbf{S}_{ij} u_j + \sum_{k \in \mathcal{N}_a(i)} \mathbf{A}_{ik} p_k$$

où $\mathcal{N}_s(i)$ est l'ensemble de tous les voisins sociaux du nœud i, (nœuds sociaux connectés au nœud i) et $\mathcal{N}_a(i)$ est l'ensemble de tous les voisins d'attribut du nœud i, (nœuds d'attribut connectés au nœud i) dans le RSA $\mathcal{G}$. Pour calculer $\mathbf{C}$, nous avons besoin de $d^2|\mathcal{N}_a(i)| + d$ opérations. Pour a, il faut $d|\mathcal{N}_s(i)| + d|\mathcal{N}_a(i)|$ opérations et pour l'affectation $u_i \leftarrow a \times \mathbf{C}^{-1}$, il faut $O(d^3)$ opérations. Une mise à jour de u_i n'a pas besoin de plus de $O(d^3 + d^2(|\mathcal{N}_s(i)| + |\mathcal{N}_a(i)|))$, où $|\mathcal{N}_s(i)| + |\mathcal{N}_a(i)|$ est le nombre de nœuds (sociaux ou d'attribut) connectés à i. Cette dernière conclusion est aussi valable pour la mise à jour des facteurs latents d'un nœud d'attribut. Dans une itération de l'algorithme, les facteurs latents de chaque nœud sont mis à jour exactement une fois. Nous en déduisons que la complexité globale de chaque itération est $O\left(d^2 N_l + d^3 N_n\right)$ où N_l et N_n sont respectivement le nombre total de liens (sociaux et attributs) et le nombre total de nœuds (sociaux et attributs) dans le RSA.

Par une analyse similaire, il est facile de voir que $O\left(d^2 N_l + d^3 N_n\right)$ est aussi la complexité pour l'approche FCM et pour l'apprentissage incrémental (rappelons que d est le nombre faible de facteurs latents que nous avons fixé). Pour l'apprentissage incrémental, N_l et N_n sont définis sur le RSA $\mathcal{G}(t)$.

Nous concluons que la complexité de notre algorithme d'apprentissage est linaire en fonction de la taille de données en entrée. Théoriquement, notre algorithme est donc capable de passer à grande échelle. Quant au nombre de facteurs latents d, la complexité de notre algorithme est d'ordre de d^3. Cela est un inconvénient de l'algorithme dans les cas où d doit prendre des valeurs relativement grandes pour avoir une bonne performance. d est petit ($d = 1$) dans nos expérimentations sur Twitter (section 4.1), mais dans les expérimentations avec les données intercanales fournies par Orange, nous avons $d = 50$. Dans nos conditions d'expérimentation sur ces dernières données, les calculs des variables sont implémentés sur Matlab (Windows 7 64 bits, CPU 2x2.27GHz). L'algorithme AIMFL prend environ 1 minute pour trouver les 50 facteurs latents (optimisation) sur un graphe attribué contenant quelques 3000 nœuds (sociaux+attributs) et 500000 liens (sociaux+attributs).

Un avantage de l'algorithme 3.1, comme la version classique de MCA pour le filtrage collaboratif [ZWSP08], est qu'il est parallélisable. La parallélisation est naturelle car rappelons que, à chaque itération, la mise à jour des facteurs latents d'un nœud ne dépend que des facteurs latents des nœuds de ses voisins dans le graphe. Les mises à jour des facteurs latents des nœuds non-connectés peuvent donc être exécutées en parallèle. L'apprentissage

à grande échelle des facteurs latents peut être effectuée sur une grande machine multi-cœur ou un cluster des machines. Dans ce travail sur les données synthétiques et Twitter, nous avons développé une version parallélisée de cet algorithme sous GraphLab [LGK+10] [1], une plateforme de calcul distribué écrite en C++. Pour $d = 20$, le calcul des facteurs latents ne dépasse pas 40 secondes pour environ 200000 nœuds sociaux, 50000 nœuds d'attribut, 350000 liens sociaux et 500000 liens d'attribut (cf. section 3.7.3).

3.7 Expérimentation avec un jeu de données synthétiques

Dans cette section, nous testons la méthode proposée avec des données synthétiques et faisons une comparaison avec d'autres méthodes. Les comparaisons sont ici surtout focalisées sur la performance. Le but de ce travail est de démontrer l'apport de notre méthode et sa capacité à détecter des facteurs latents informatifs dans des données intercanales, cela afin de prédire une variable cible. Nous décrivons d'abord notre générateur de données synthétiques.

3.7.1 Le générateur des données synthétiques

Notre but est de générer un jeu de données conforme au contexte de la thèse : celui du CRM multicanal. Nous simulons ici deux canaux : les média sociaux et la base de données clientèle que nous désignons par le SI client (*système d'information client*). Les données des média sociaux sont générées comme un RSA daté (avec une date de création sur chaque lien). Dans ce graphe augmenté, chaque nœud social correspond à un client et chaque nœud d'attribut correspond à un type de contenu créé par les clients sur les média sociaux, par exemple les mots qu'ils écrivent dans leur posts sur les média sociaux. À chaque pas de temps t, nous pouvons avoir des nouveaux nœuds sociaux et des nouveaux nœuds d'attribut. Les données tabulaires issues du SI client sont générées à chaque pas de temps de manière *indépendante* [2] des données issues des média sociaux. Il s'agit des variables explicatives des clients dans la base de données clientèle.

Ensuite, à chaque pas de temps nous générons une variable cible qui dépend à la fois du graphe social, des attributs sociaux et des attributs SI client. Cette variable cible correspond

1. GraphLab est un projet open-source et disponible à l'adresse http ://graphlab.org/

2. Notons ici que nous faisons une hypothèse forte d'indépendance des données provenant de ces deux sources. En réalité, l'indépendance n'est pas avérée, et notre intuition est plutôt contraire. Notre première préoccupation est d'utiliser conjointement les données des média sociaux et celles du SI pour prédire une variable cible ; nous avons donc besoin de générer une variable cible qui dépend de ces deux sources de données. Nous avons laissé de côté les dépendances entre ces deux sources, qui ne sont pas évidentes à simuler. Nous avons donc fait le choix de générer des données type SI client indépendantes des données sociales, et nous contrôlons par contre la dépendance de la variable cible avec les deux sources de données.

à un acte commercial, par exemple le changement d'offre, que nous voulons prédire. Nous nous sommes donc basés sur l'hypothèse suivante : les actes commerciaux (futurs) d'un client dépend à la fois du graphe social (avec qui le client a interagi sur les média sociaux), les attributs sur les médias sociaux (e.g les posts qu'il a écrit) et les variables caractérisant le client dans le SI client.

Notre générateur de données synthétiques contient donc les composants suivants :

- Le générateur de RSA représentant les données issues des média sociaux
- Le générateur de variables explicatives dans le SI client
- Le générateur de variable cible

3.7.1.1 Le générateur de RSA représentant les données des média sociaux

Notre point de départ : le modèle de Gong et al. [GXH$^+$12] Nous sommes inspirés du générateur de RSA proposé par Gong et al. [GXH$^+$12]. Basé sur les observations empiriques de création du réseau social attribué Google+, ils ont proposé un nouveau modèle génératif pour reproduire conjointement la structure sociale et les attributs de Google+. Le générateur de RSA proposé dans [GXH$^+$12] est décrit dans l'algorithme 3.2.

Algorithm 3.2 L'algorithme pour le modèle génératif de Gong et al. [GXH$^+$12]

Initialisation
pour $0 \leq t \leq T$ **faire**
  Échantillonner l'ensemble de $N(t)$ nouveaux nœuds sociaux $V_{t,nouveau}$
  pour tous $v_{nouveau} \in V_{t,nouveau}$ **faire**
    Échantillonner le degré d'attributs n_a pour $v_{nouveau}$ selon une loi log-normale
    pour $0 \leq j \leq n_a$ **faire**
      Relier $v_{nouveau}$ vers les attributs
    fin pour
    Relier $v_{nouveau}$ vers un nœud social existant (premier lien social)
    Échantillonner la duré de vie
    Échantillonner le temps de sommeil
  fin pour
  Identifier l'ensemble des nœuds réveillés $V_{t,rveill}$
  pour tous $v_{reveille} \in V_{t,rveill}$ **faire**
    Relier $v_{reveille}$ vers un nœud social existant (fermeture de triangle)
    Échantillonner le temps de sommeil
  fin pour
fin pour

Pour le générateur de [GXH+12], le graphe social généré est dirigé (les liens sont orientés). Nous ne nous intéressons pas à l'orientation de liens sociaux parce que notre algorithme est appliqué sur un graphe social non-orienté. Nous allons simplement traiter les liens orientés comme des liens non-orientés et dans la description qui suit, nous allons oublier l'orientation des liens. Par exemple, nous ne différencierons pas le *degré entrant* et le *degré sortant* comme dans le texte original [GXH+12]. Les nœuds sociaux arrivent à un taux prédéterminé. À l'arrivée, chaque nœud social prend son ensemble d'attributs et connecte à son premier voisin social. Après avoir rejoint le réseau, chaque nœud « dort » pendant un certain temps, se réveille et ajoute de nouveaux liens. Voici une description brève de chaque composant dans l'algorithme 3.2 :

Initialisation (ligne 1) Dans cette étape le graphe attribué est initialisé avec un certain nombre de nœuds sociaux et de nœuds d'attribut. Il est observé que le choix de ces nombres n'a pas un grand impact sur la structure du RSA généré.

Arrivée des nœuds sociaux (ligne 3) Les nœuds sociaux arrivent selon une fonction $N(t)$. Dans leurs simulations, Gong et al. [GXH+12] laissent tout simplement $N(t) = 1$ pour modéliser l'arrivée de chaque nœud comme pas de temps discret.

Le degré d'attribut (ligne 5) Chaque nœud social prend un certain nombre d'attributs n_a échantillonné à partir d'une loi log-normale avec la moyenne μ_a et la variance σ_a^2.

Création de liens d'attribut (ligne 7) À son arrivée, chaque nœud social se connecte à n_a nœuds d'attribut. Pour chaque attribut, avec une probabilité p_a, un nouveau nœud d'attribut a est généré ; sinon un nœud d'attribut existant est choisi avec une probabilité proportionnelle à son degré social (i.e le nombre de liens d'attribut). Il s'agit donc d'un modèle d'*attachement préférentiel* d'attribut (le modèle d'attachement préférentiel [BA99] est souvent utilisé dans les générateurs de réseaux sociaux. Le principe est qu'un nouveau nœud se connecte à un nœud existant avec une probabilité proportionnelle à son degré).

Premier lien social (ligne 9) Chaque nouveau nœud émet un lien social vers un nœud social selon le modèle d'attachement préférentiel augmenté. Ce modèle (*Linear Attribute Preferential Attachment* ou *LAPA*) choisit un nœud social existant avec une probabilité proportionnelle à une combinaison du degré social et du nombre de voisins communs entre deux nœuds sociaux. Plus précisément :

$$f(u, v_{nouveau}) \propto d(u)^{\alpha}(1 + \beta \cdot a(u, v_{nouveau}))$$

où $f(u, v_{nouveau})$ est la probabilité de $v_{nouveau}$ de se lier à u comme premier voisin, $d(u)$ est le degré social de u, $a(u, v_{nouveau})$ est le nombre de voisins communs entre u et $v_{nouveau}$.

Durée de vie (ligne 10) La durée de vie l de $v_{nouveau}$ est échantillonnée à partir d'une loi normale tronquée, i.e $l \propto \exp(-\frac{(l-\mu_l)^2}{2\sigma_l^2})$

Durée de sommeil (lignes 11 et 16) Après avoir rejoint le réseau, tissé des liens, ou après un pas de temps réveillé, chaque nœud ensuite « dort » pendant un certain temps. La durée de sommeil est échantillonnée selon n'importe quelle distribution avec une moyenne de $\frac{m_s}{d(u)}$, où m_s est un paramètre de modèle génératif et $d(u)$ est le degré social du nœud u. L'intuition de prendre la durée de sommeil inversement proportionnelle au degré sortant est qu'un nœud avec un plus grand degré a plus tendance à émettre des liens.

Fermeture de triangle Chaque nœud social réveillé $v_{reveille}$ délivre un nouveau lien selon le modèle de fermeture de triangle appelé *Random-Random Social Attribute Network* (RR-SAN) : on sélectionne aléatoirement un voisin w de $v_{reveille}$ (w est un nœud social ou un nœud d'attribut) et ensuite on sélectionne aléatoirement un voisin social v de w, le nœud $v_{reveille}$ émet un lien social vers v.

Gong et al. [GXH+12] ont montré que ce modèle génératif est capable de générer un RSA qui est proche du RSA réel de Google+ en termes de structure du graphe social et des attributs (via des métriques comme la densité, la distribution de degré, le coefficient de clustering, etc.) et en termes d'évolution du graphe.

Cet algorithme complexe a besoin de beaucoup de paramètres à régler afin de reproduire un RSA proche de Google+. Ce n'est pas notre préoccupation dans ce travail. Pour tester notre méthode, nous voulons seulement créer un RSA dynamique (avec une date de création sur chaque lien) dans lequel la structure du graphe social co-évolue avec les attributs sociaux. De plus, nous voyons que le modèle de l'algorithme 3.2 se concentre actuellement sur les attributs statiques, dans le sens où un nœud social se lie avec des nœuds d'attribut lorsqu'ils rejoignent le RSA et son ensemble d'attributs ne change pas après. Nous devons donc apporter quelques modifications pour concevoir notre générateur de RSA. L'idée est de réutiliser et adapter quelques composants du générateur de Gong et al. [GXH+12] pour créer notre propre générateur avec différents types d'attribut, et avec une évolution des liens d'attribut notamment.

Notre modèle génératif Notre modèle génératif pour le RSA représentant les données issues de média sociaux, inspiré du travail de Gong et al. [GXH+12], est décrit dans l'algorithme 3.3.

Voici les modifications principales que nous avons apportées :

– Gong et al. [GXH+12] a modélisé l'arrivée de chaque nœud comme pas de temps discret. Comme notre modèle considère un pas de temps comme un instant réel, dans chaque pas de temps il y a plusieurs nouveaux nœuds sociaux ($N(t) > 1$). Ce nou-

Algorithm 3.3 L'algorithme modifié pour générer un RSA représentant les données issues des média sociaux

1: Initialisation
2: **pour** $0 \leq t \leq T$ **faire**
3: Échantillonner l'ensemble de $N(t)$ nouveaux nœuds sociaux $V_{t,nouveau}$
4: **pour tous** $v \in V_{t,nouveau}$ **faire**
5: Relier v vers un nœud social existant (premier lien social)
6: **fin pour**
7: $V_t = V_{t,nouveau} \cup V_t$
8: **pour tous** $v \in V_t$ **faire**
9: Échantillonner le nombre d'attributs n_a pour v selon une loi log-normale
10: **pour** $0 \leq j \leq n_a$ **faire**
11: Relier v vers les attributs
12: **fin pour**
13: Relier v vers un nœud social existant (fermeture de triangle)
14: **fin pour**
15: **fin pour**

veaux nœuds correspondent aux nouveaux clients ou nouveaux participants des média sociaux que nous observons dans la période entre deux pas de temps.

– Nous avons enlevé le mécanisme de « dormir » et « se réveiller » concernant des nœuds sociaux. Dans notre modélisation, comme un pas de temps correspond à une instant (par exemple, début de semaine), nous supposons que les acteurs sociaux restent actifs (i.e au moins une fois ils tissent des liens sociaux ou des liens d'attribut) dans tous les pas de temps. Cette modification d'une part simplifie le modèle, et d'autre part et surtout, s'adapte bien à notre besoin.
– Dans notre modèle, un nœud social peut se relier à des attributs à tout moment. Cela donne une certaine dynamique sur des attributs des nœuds sociaux, contrairement au modèle original de Gong et al. [GXH+12] dans lequel un nœud social prend des attributs seulement une fois lors de son arrivée.

En dehors de ces modifications, notre modèle reprend les composants du modèle de base décrit dans l'algorithme 3.2, avec quelques petites adaptations. Voici une description des composants des composants de notre modèle génératif

Initialisation (ligne 1) Comme dans l'algorithme 3.2 nous initialisons le graphe avec quelques nœuds sociaux et quelques nœuds d'attribut (dans notre expérimentation nous fixons à 5 nœuds sociaux, 5 nœuds d'attribut au départ).

Arrivée des nœuds sociaux (ligne 3) Les nœuds sociaux arrivent selon une fonction $N(t)$. Comme décrite ci-dessus, dans notre modèle, $N(t)$ correspond à un ensemble de nouveaux clients à t.

Premier lien social (ligne 5) Nous reprenons le modèle d'attachement préférentiel de l'al-

gorithme 3.2.

Le nombre d'attributs sur chaque nœud (ligne 9) . Nous réutilisons le composant correspondant de l'algorithme 3.2 ; à chaque pas de temps, chaque nœud social se lie à n_a nœuds d'attribut, n_a suit une loi log-normale de paramètre μ_a, σ_a.

Création de liens d'attribut (ligne 13) Chaque nœud social v se lie à n_a nœud d'attribut. De façon similaire à l'algorithme 3.2, pour chaque attribut, avec une probabilité p_a un nouveau nœud d'attribut a est généré ; sinon un nœud d'attribut existant est choisi. Dans l'algorithme 3.2, un nœud d'attribut existant est choisi selon le modèle d'attachement préférentiel. Ici, nous modifions la manière de choisir un attribut existant pour renforcer la corrélation entre les attributs et la structure du graphe social : d'abord, on sélectionne aléatoirement un voisin social u de v et ensuite on prend aléatoirement un nœud d'attribut parmi les nœuds d'attribut connectés à u. De cette manière, les voisins sociaux sont plus probables d'avoir les mêmes attributs. Dans les média sociaux, l'intuition derrière cette modélisation est l'hypothèse d'*homophilie* : qui se ressemble s'assemble, ici les gens connectés dans le graphe social tendent à parler des mêmes choses (e.g. écrire les mêmes mots).

Fermeture de triangle Nous reprenons ce composant de l'algorithme 3.2.

L'algorithme modifié décrit ci-dessus permet de générer un RSA dans lequel les liens sont datés (par t, le pas de temps). À chaque pas de temps, nous avons des nouveaux nœuds sociaux, des nouveaux nœuds d'attribut, et les liens vers les attributs co-évoluent avec la structure sociale au fil du temps, ce qui répond à notre besoin.

Nous résumons les paramètres de notre modèle génératif et leurs implications sur les données générées :

- T, le nombre de pas de temps (périodes) simulés
- $N(t), t = 1, 2, ..., T$, le nombre de nouveaux nœuds sociaux dans chaque pas de temps
- μ_a, σ_a, les paramètres de la distribution log-normale du nombre d'attributs qu'un nœud social prend dans pas de temps. Ces paramètres contrôlent le nombre liens d'attribut créés dans chaque pas de temps.
- p_a, la probabilité de créer un nouveau nœud d'attribut à chaque fois que l'on cherche à connecter un nœud social vers un nœud d'attribut. Ce paramètre contrôle le taux de création de nouveaux attributs.

3.7.1.2 Le générateur de données tabulaires du SI client

Le deuxième composant de notre générateur de données synthétiques est le générateur de données du SI client. Nous allons intégrer ce composant dans l'algorithme 3.3. L'idée est de générer les variables explicatives du SI sur les nœuds sociaux dans chaque pas de temps.

Dans ce travail, nous modélisons deux variables issues du SI client : la première variable, notée par *consom*, représente la *consommation* des clients (par exemple, avec les clients mobile, la consommation correspond au nombre d'appels ou durée totale d'appel) ; la deuxième variable, notée par *duree_restante*, est la durée restante du contrat des clients (nous supposons que tous les clients signent un contrat à durée déterminée). Les valeurs de ces variables explicatives sont générées sur chaque client (nœud social) dans chaque pas de temps, de manière indépendante du RSA représentant les données issues des média sociaux.

Pour la variable *consom*, nous modélisons trois modalités : consommation élevée, consommation moyenne et consommation faible. À chaque pas de temps, pour chaque client, on choisit une de trois modalités pour chaque client selon une loi discrète de paramètre (c_{elevee}, $c_{moyenne}$, c_{faible}).

Pour la variable *duree_restante*, nous voulons modéliser la durée restante (mesurée en nombre de pas de temps dans notre modélisation) jusqu'à la fin du contrat du client. À chaque pas de temps, la valeur de cette variable est générée comme suit :

- Pour un nouveau nœud social, la valeur *duree_restante* est tirée uniformément entre 1 et D_{max}, où D_{max} représente la durée maximale de contrat d'un client.
- Pour un nœud social existant dont *duree_restante* est positive, on va diminuer cette valeur par 1.
- Pour un nœud social existant dont $duree\_restante = 0$ (la fin du contrat), la valeur *duree_restante* est réinitialisée en tirant uniformément une valeur entre 1 et D_{max} (on considère que le contrat est « renouvelé », et la durée de renouvellement n'est pas la même pour tout le monde).

Le composant de générateur de données du SI a donc comme paramètres : c_{elevee}, $c_{moyenne}$, c_{faible} (pour la variable *consom*), D_{max} pour la variable *duree_restante*.

3.7.1.3 Le générateur de variable cible

Le dernier composant de notre générateur est le générateur de variable cible. Il s'agit d'une variable binaire (positive ou négative) indiquant si un client effectue un acte commercial particulier. Nous appelons aussi une valeur positive de la variable cible une étiquette positive [1].

Comme mentionné avant, cette variable dépend à la fois du graphe social, des attributs sociaux et des variables du SI client. À chaque pas de temps, pour un nœud social v nous calculons un *score*, noté par $score_t(v)$, qui correspond à la probabilité que le client effectuera l'acte commercial à $t+1$ (i.e la valeur de la variable cible sera positive à $t+1$). Ce *score* est

1. Nous considérons le type d'acte commercial qu'un client ne peut effectuer qu'une fois (par exemple, pour les clients mobile, la migration de contrats)

une combinaison linaire de trois composants :

$$score_t(v) = \alpha_{attri} \cdot score\_attri_t(v) + \alpha_{soc} \cdot score\_social_t(v) + \alpha_{si} \cdot score\_si_t(v) \tag{3.21}$$

Les trois termes de cette combinaison correspondent aux dépendances de la variable cible avec le graphe social, les attributs sociaux et les variables du SI client, respectivement. Ces trois termes sont pondérés par des paramètres α_{attri}, α_{soc} et α_{si} qui permettent de régler la contribution de chaque partie dans le *score* total. Après avoir calculé ce *score* pour les nœuds, on sélectionne les nœuds ayant les scores les plus élévés (ϵ parmi ceux qui n'ont pas encore une valeur positive de la variable cible avant t) et on affecte une étiquette positive pour ces nœuds. Autrement dit, les nœuds ayant les tops scores correspondent aux clients qui adopteront l'acte commercial à $t+1$.

Dans la suite nous décrivons chaque composant de dépendance en détail.

score_attri$_t(v)$ Ce facteur représente la dépendance de la variable cible $y_t(v)$ avec les nœuds d'attribut issus des média sociaux. Chaque nœud d'attribut connecté à nœud social peut avoir un impact positif (faire augmenter) ou négatif (faire diminuer) la probabilité d'avoir une étiquette positive sur ce nœud social. Nous modélisons donc l'impact de chaque attribut sur la variable cible par un nombre réel dans l'intervalle $[-1, 1]$. Dans notre générateur, le facteur d'impact de l'attribut k, noté par f_k, est généré uniformément dans l'intervalle $[-1, 1]$ lors de l'arrivée du nœud d'attribut. L'impact total de tous les attributs sociaux connectés à un nœud social v, est calculé comme suit :

$$score\_attri_t(v) = \sum_{k \in \mathcal{N}_a^{\leq t}(v)} f_k \tag{3.22}$$

où $\mathcal{N}_a^{\leq t}(v)$ est l'ensemble des nœuds d'attribut connectés à v jusqu'au pas de temps t. La modélisation de *score_attri*$_t(v)$ est basée sur une intuition très simple : considérons les attributs tels que les mots qu'un client écrit sur les média sociaux. Chaque mot exprime plus ou moins une opinion du client sur l'acte commercial (positive, négative ou neutre), et donc il a un impact particulier sur la valeur de la variable cible. Les impacts des mots sont indépendants les uns des autres, et l'impact total est simplement la somme des impacts de tous les mots.

score_soc$_t(v)$ Ce facteur représente la dépendance de la variable cible avec le graphe social. Nous nous sommes basés sur l'hypothèse d'influence sociale : la décision d'un client est influencée par ses voisins dans le graphe social. Dans ce modèle, la probabilité qu'un nœud social prenne une étiquette positive est proportionnelle à la portion de

ses voisins en ayant une. Formellement, $score\_soc_t(v)$ est défini comme suit :

$$score\_soc_t(v) = \frac{\sum_{u \in \mathcal{N}_s^{\leq t}(v)} \mathbb{1}\{y_{t'}(u) = 1, t' \leq t\}}{\left|\mathcal{N}_s^{\leq t}(v)\right|} \tag{3.23}$$

où $\mathcal{N}_s^{\leq t}(v)$ est l'ensemble de nœuds sociaux voisins de v jusqu'à t, $y_{t'}(v)$ la valeur de la variable cible du nœud v au pas de temps t' (qui indique si un client a effectué l'acte commercial dans la période entre deux instant t' et $t\prime + 1$). ,$\mathbb{1}\{y_{t'}(u) = 1, t' \leq t\}$ est une fonction d'indicateur qui vaut 1 si u a une étiquette positive avant t.

score_si$_t$ Ce facteur représente la dépendance de la variable cible avec les variables explicatives du SI. Nous avons deux variables explicatives : *consom* et *duree_restante*. Les impacts de ces variables sur la variable cible sont comme suit :

- Pour la variable *consom*, si cette variable a une valeur moyenne, elle n'a pas d'impact sur la probabilité d'effectuer l'acte commercial. En revanche, lorsque cette variable prend une valeur faible ou élevée, on augmente cette probabilité. Voici les idées derrière cette modélisation : lorsque l'on a des problèmes (mauvaise réception, terminal en panne) ou de faibles besoins (indiqué par une valeur faible de consommation), on peut avoir envie de partir ou de changer d'offre ; lorsque l'on consomme beaucoup plus que la moyenne, l'offre que l'on a n'est peut-être plus adaptée et on peut avoir besoin ou envie de changer d'offre.
- Pour la variable *duree_restante*, nous supposons que la probabilité qu'un client adopte un acte commercial augmente lorsque sa fin du contrat approche. Cela correspond aux situations réelles où les clients souvent changent d'offres ou d'opérateur, ou encore achètent un nouveau terminal, vers la fin du contrat, pour éviter des pénalisations ou avoir des récompenses. Les impacts des deux variables du SI sont résumés dans l'équation suivante :

$$\begin{aligned} score\_si_t(v) =& \mathbb{1}\{consom_t(v) = elevee \vee consom_t(v) = faible\} \\ &+ \mathbb{1}\{duree\_restante_t(v) = 0\} \end{aligned} \tag{3.24}$$

où $\mathbb{1}\{consom_t(v) = elevee \vee consom_t(v) = faible\}$ est une fonction d'indicateur qui vaut 1 si la consommation de v est élevée ou faible à t et $\mathbb{1}\{duree\_restante_t(v) = 0\}$ est une fonction d'indicateur qui vaut 1 si la durée restante du contrat de v est 0 à t.

Dans l'algorithme 3.4, nous résumons le fonctionnement de notre générateur de données synthétiques. L'algorithme 3.4 reprend la base de l'algorithme 3.3 avec deux composants ajoutés : le générateur des variables du SI (ligne 15 à 22) et le générateur de la variable cible (ligne 23 à 27). Nous listons ici les paramètres de ces nouveaux composants :

- c_{eleve}, $c_{moyenne}$, c_{faible} : les paramètres pour générer la variable *consom*, qui définissent les proportions des clients ayant une consommation élevée, moyenne et faible.

- D_{max} : le paramètre pour générer la variable *duree_restante*, qui correspond à la durée maximale du contrat des clients.
- α_{attri}, α_{soc} et α_{si} : les paramètres permettant de régler les contribution de chaque type d'information (graphe social, attributs sociaux ou variables du SI) dans la modélisation de la variable cible.

Algorithm 3.4 L'algorithme du générateur de données synthétiques

Initialisation
pour $0 \leq t \leq T$ **faire**
  Échantillonner l'ensemble de $N(t)$ nouveaux nœuds sociaux $V_{t,nouveau}$
  pour tous $v \in V_{t,nouveau}$ **faire**
    Relier v vers un nœud social existant (premier lien social)
  fin pour
  $V_t = V_{t,nouveau} \cup V_t$
  pour tous $v \in V_t$ **faire**
    Échantillonner le nombre d'attributs n_a pour v selon une loi log-normale
    pour $0 \leq j \leq n_a$ **faire**
      Relier v vers les attributs
    fin pour
    Relier v vers un nœud social existant (fermeture de triangle)
  fin pour
  pour tous $v \in V_t$ **faire**
    Échantillonner une valeur pour $consom_t(v)$, selon une loi discrète de paramètre $(c_{eleve}, c_{moyenne}, c_{faible})$
    si $v \in V_{t,nouveau}$ ou $duree\_restante_{t-1}(v) = 0$ **alors**
      Échantillonner une valeur pour $duree\_restante_t(v)$, uniformément parmi les valeurs entre 1 et D_{max}
    sinon
      $duree\_restante_t(v) = duree\_restante_{t-1}(v) - 1$
    fin si
  fin pour
  Collecter les nœuds V_t^0 qui n'ont pas encore une étiquette positive avant t
  pour tous $v \in V_t^0$ **faire**
    Calculer $score_t(v)$ selon les équations 3.21, 3.22, 3.23 et 3.24
  fin pour
  Sélectionner $\epsilon \cdot \|V_t^0\|$ nœuds (parmi V_t^0) ayant les valeurs de $score_t$ les plus élevées et attribuer une étiquette positive à ces nœuds
fin pour

3.7.2 Expérimentation

Après avoir conçu le générateur de données synthétiques, nous menons des expérimentations : générer un premier jeu de données, y appliquer notre méthode et les autres méthodes et comparer les performances. Nous étudions aussi les impacts des paramètres de notre méthode. Finalement, nous testons d'autres scénarios pour observer le comportement de notre méthode dans une variété de situations.

3.7.2.1 Le premier scénario

Génération du jeu de données Pour un premier test de notre méthode, nous avons généré un jeu de données par notre générateur décrit dans les paragraphes précédents, avec la configuration suivante :

- Nous simulons 10 pas de temps ($T = 10$). Au premier pas de temps, $t = 0$, nous avons 100000 nœuds sociaux ($N(0) = 100000$), ensuite on rajoute 10000 nœuds sociaux à chaque pas de temps ($N(t) = 10000, t = 1, 2, ..., 9$. Nous voulons simuler la situation suivante en réalité : nous avons déjà un nombre assez important de clients dans notre base de données (100000 clients) et à chaque pas de temps, nous ajoutons une petite portion de clients (10000 à chaque fois).
- Pour les données issues des média sociaux, les paramètres du modèle génératif des données sociales sont choisis comme suit : $p_a = 0,1$ (avec une probabilité 0,1 un nouvel attribut sera créé chaque fois qu'on crée un lien d'attribut), $\mu_a = 1, \sigma_a = 1$ (à chaque pas de temps, le nombre de liens d'attribut créés pour un nœud social suit la loi log-normale $\mu_a = 1, \sigma_a = 1$).
- Pour les variables du SI, les paramètres sont choisis comme suit : $c_{eleve} = c_{faible} = 0.05$, $c_{moyenne} = 0.9$ (c'est-à-dire, 90% des clients ont une consommation normale, 5% consomment beaucoup et 5% consomment très peu par rapport à la moyenne). La durée maximale du contrat est $D_{max} = 20$.
- Pour la variable cible, les paramètres sont $\alpha_{attri} = \alpha_{soc} = \alpha_{si} = 1.0$. Tous les trois éléments dans les données : le graphe social, les attributs et les variables explicatives (SI client) ont des impacts sur la variable cible. La portion de population ayant une étiquette positive (ϵ) est fixée à 0.1. Nous voulons simuler les actes commerciaux effectués par une petite proportion des clients (10% à chaque pas de temps).

Les statistiques sur le jeu de données synthétiques généré sont présentées dans l'annexe A.

Application de notre méthode Pour appliquer notre méthode basée sur l'apprentissage incrémental des facteurs latents, nous transformons d'abord les données synthétiques en forme d'une séquence de RSAs. La construction des RSAs à chaque pas de temps est décrite

dans la section 3.3.3. Ici, la transformation de la partie « média sociaux » de données synthétiques vers la forme de RSA est directe. Pour transformer les variables du SI sous forme de RSA, nous créons un nœud d'attribut pour chaque modalité des variables. Nous avons donc trois nœuds d'attribut pour la variable *consom* qui correspondent aux trois modalités de cette variable : consommation élevée, consommation moyenne, consommation faible. Pour la variable *duree_restante*, nous créons deux nœuds d'attribut correspondant aux deux situations : un client est à la fin de son contrat ou pas.

Une fois que nous avons obtenu une séquence de RSAs représentant les données intercanales, nous appliquons notre méthode (apprendre les facteurs latents sur les individus et utiliser les facteurs latents comme variables explicatives) pour prédire la variable cible à chaque pas de temps $t, t = 1, ..., 9$. Dans un premier temps, les paramètres de notre méthode sont choisis comme suit : $\lambda = 1.0, \alpha = 1.0, \mu = 1.0$ et le nombre de facteurs latents $d = 20$. Les impacts de ces paramètres sont étudiés dans une section suivante.

Les méthodes de références Notre méthode est comparée avec les autres méthodes de prédiction en termes de performances. Les méthodes de référence que nous avons implémentées pour comparer avec notre méthode sont :

- *svm attri*. Nous utilisons la classification supervisée SVM avec les attributs dans la partie « média sociaux » de données. À chaque pas de temps t, nous apprenons un classifieur SVM. Dans le pas de temps suivant $t + 1$ nous utilisons le modèle SVM appris à t pour prédire la variable cible. Nous utilisons l'implémentation SVM de Fan et al. [FCH+08] avec le noyau linaire. Cette méthode de prédiction utilise seulement les attributs sur les nœuds sociaux, elle correspond au composant du générateur de la variable cible décrit dans l'équation 3.22
- *svm si*. Nous utilisons la classification supervisée SVM avec les variables du SI de la même manière qu'avec les attributs sociaux : au pas de temps t, nous apprenons un classifieur SVM et au pas de temps suivant $t + 1$ nous déployons le classifieur pour prédire la variable cible. Cette méthode de prédiction utilise seulement les variables du SI, elle correspond au composant du générateur de la variable cible décrit dans l'équation 3.24
- *méthode de voisinage*. Cette méthode utilise seulement le graphe social. Elle correspond à une approche de fouille de graphe que nous avons révisé en chapitre 2 (les modèle de l'influence - section 2.2). À chaque pas de temps t, nous construisons un graphe social qui contient tous les liens sociaux datés jusqu'à t. Ensuite, nous calculons le score de prédiction pour chaque individu dans ce graphe. Au pas de temps t, le score d'un individu est égal à la proportion de ses voisins ayant une étiquette positive (i.e ceux qui ont adopté l'acte commercial avant t). Nous voyons bien que ce prédicteur

n'a besoin d'aucun apprentissage et correspond au composant du générateur de la variable cible décrit dans l'équation 3.23 (i.e l'impact du graphe social sur la variable cible).

– *svm sociodim*. Cette méthode utilise également seulement le graphe social. Elle est une autre approche de l'utilisation du graphe social pour la classification proposée par [TL11]. Nous avons décrit cette approche dans la section **??**.Dans cette expérimentation, nous adaptons cette approche pour notre problème de prédiction : extraire les dimensions sociales à chaque pas de temps, apprendre un classifier avec les dimensions sociales au pas de temps t et le déployer au pas de temps $t + 1$. Le nombre de dimensions sociales (i.e nombre de clusters) est fixé à 20 comme nous n'avons observé aucune amélioration en termes performances en augmentant ce paramètre au delà de 20.
– *svm sociodim+attri+si*. Il s'agit d'une combinaison de *svm attri*, *svm si* et *svm sociodim*. Nous utilisons l'apprentissage supervisé (SVM avec le noyau linaire) avec comme variables explicatives : les attributs sociaux, les variables du SI et les dimensions sociales. Cette combinaison permet d'utiliser toutes les informations prédictives dans les données générées.

Remarque Par faute de temps, nous n'avons pas implémenté la méthode basée sur la construction des variables explicatives à partir des interactions sociales à chaque pas de temps (cette méthode consiste à calculer les variables explicatives sur les individus à chaque pas de temps, puis à les utiliser pour l'apprentissage supervisé) sur ces données synthétiques. Néanmoins, nous l'avons implémentée pour les expérimentations sur les données réelles (chapitre 4).

Les performances Nous utilisons l'aire sous la courbe ROC (*Area Under Reciever Operating Characteristic Curve - AUC*) [Bra97] pour mesurer la performance de prédiction. Toutes les méthodes de prédiction listées ci-dessus donne un *score* de prédiction à chaque individu, qui correspond à la probabilité d'avoir une étiquette positive ; il faut donc choisir un seuil de coupure pour avoir des prédictions finales (positif ou négatif). AUC est plus adapté comme mesure de performance dans ce cas : elle permet de mesurer la performance de prédiction dans tous les seuils de coupure possibles. L'avantage d'utiliser l'AUC est que nous n'avons pas besoin de fixer un seuil de coupure pour chaque méthode. Par rapport aux autres mesures (comme par exemple la précision ou le rappel), cette mesure est une bonne mesure de performance pour comparer différentes méthodes, même si la répartition des exemples positifs et négatifs dans nos données est inégale. Un AUC de 0.5 correspond en général au hasard (un modèle qui donne des prédictions aléatoires), un AUC de 1 serait un modèle

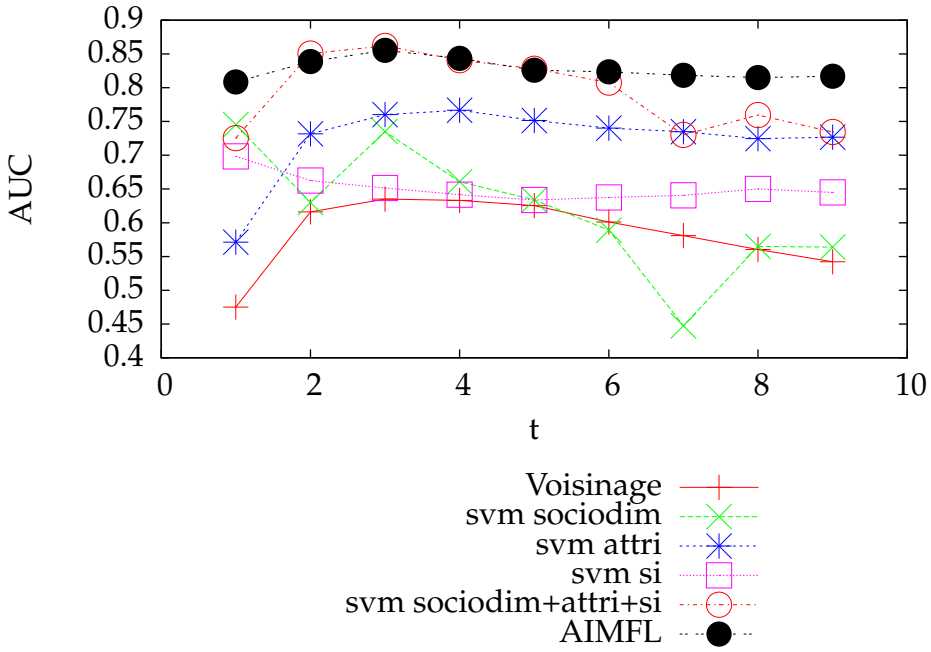


FIGURE 3.5 – Performances (AUC) des différentes méthodes

parfait (qui ne fait jamais aucune erreur quelque soit le seuil).

Les performances de prédictions des différentes méthodes, en termes d'AUC, sont présentées dans la figure 3.5.

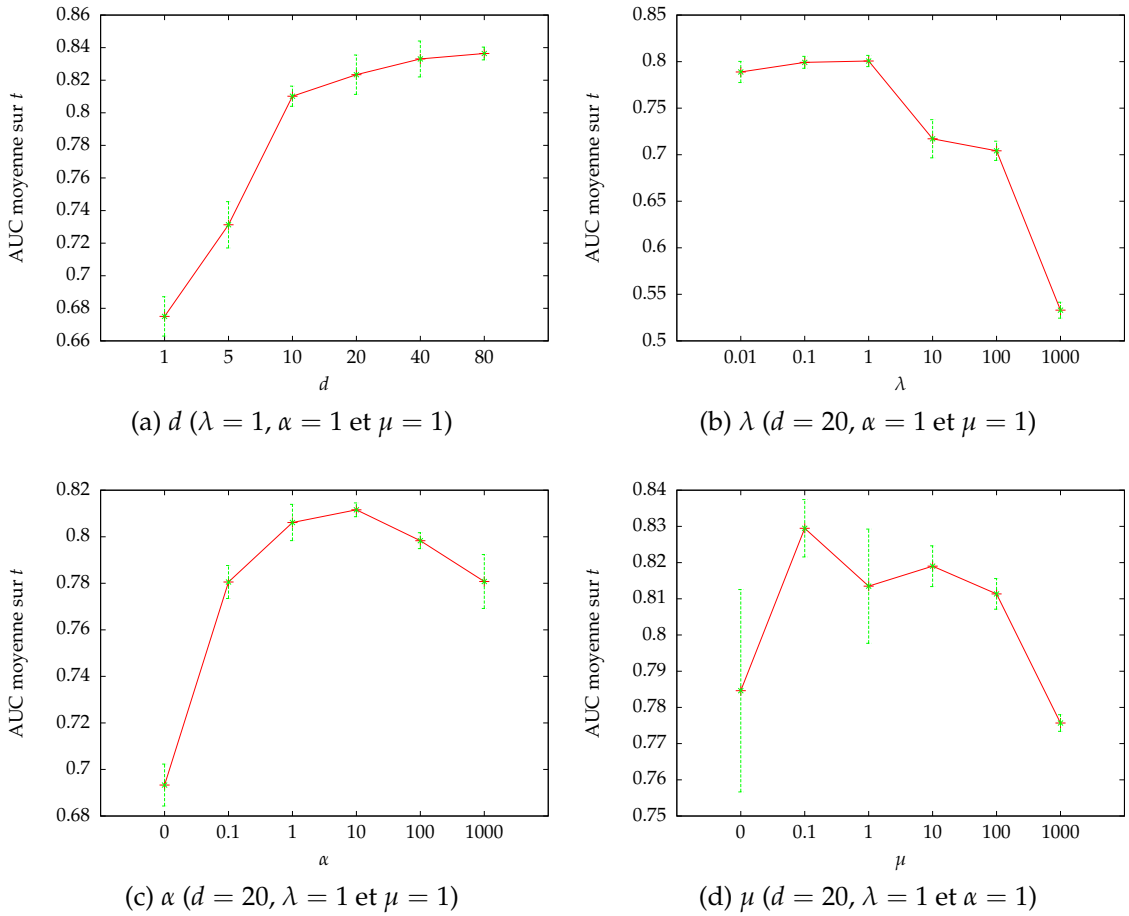
Nous remarquons d'abord que les 3 types d'informations, attributs sociaux, graphe social et variables du SI sont informatifs pour prédire la variable cible. L'utilisation de l'un de ces trois type d'information (i.e les méthodes *svm attri, svm si et voisinage*) permet de prédire la variable cible avec une AUC variable entre 0.55 et 0.75. La méthode *svm sociodim+attri+si*, qui combine tous les types d'information, donne la meilleure performance (AUC entre 0.7 et 0.8). Cela est un résultat attendu : notre variable cible est une combinaison linaire des attributs sociaux, du graphe social et les variables du SI ; l'utilisation des trois types d'information pour la prédiction donne donc la meilleure performance.

La méthode *svm sociodim* donne des meilleures performances que la méthode de *voisinage* dans certains pas de temps. L'inconvénient de cette méthode est qu'elle n'est pas stable : l'AUC varie beaucoup d'un pas de temps à l'autre (et l'AUC à $t = 7$ est inférieure à 0.5). Ce phénomène peut être expliqué comme suit. La méthode de *svm sociodim* utilise les dimensions sociales (i.e les appartenances des nœuds aux clusters du graphe social) comme variables explicatives ; à chaque pas de temps t on apprend un modèle et on le déploie au pas de temps suivant $t + 1$. Comme il y a des nouveaux nœuds et liens dans le graphes sociaux ajoutés à chaque pas de temps, la structure des clusters dans le graphe peut changer d'un pas de temps à l'autre ; l'espace « dimensions sociales » peut changer. En d'autres

termes, il est probable que les dimensions sociales au pas de temps $t+1$ ne signifient pas les mêmes sémantiques que celles au pas de temps t. Quand on apprend un modèle avec les dimensions sociales à t on le déploie avec les dimensions sociales à $t+1$, la performance de prédiction peut être influencée par ce changement.

Notre méthode de prédiction (AIMFL), basée sur les facteurs latents, donne une performance comparable avec la méthode *svm sociodim+attri+si*. La performance de prédiction est stable (reste au même niveau autour de 0.8) dans tous les pas de temps. Notre méthode d'apprentissage des variables latents a réussi à extraire des variables latentes informatives à partir des données. En d'autres termes, par un modèle à variables latentes, nous avons réussi à exploiter simultanément différents types d'informations dans les données intercanales : le graphe social, les attributs sur les acteurs sociaux et les variables du SI.

3.7.2.2 Les impacts des paramètres sur la performance



(a) d ($\lambda = 1$, $\alpha = 1$ et $\mu = 1$)

(b) λ ($d = 20$, $\alpha = 1$ et $\mu = 1$)

(c) α ($d = 20$, $\lambda = 1$ et $\mu = 1$)

(d) μ ($d = 20$, $\lambda = 1$ et $\alpha = 1$)

FIGURE 3.6 – Effet des paramètres de la méthode AIMFL

Dans cette section, nous examinons les effets des paramètres de notre méthode de prédiction sur la performance. Nous utilisons le jeu de données généré dans le premier scé-

nario (décrit au-dessus) dans ces expérimentations. L'idée est de faire varier un paramètre en fixant les autres pour voir son impact sur la performance de prédiction. Notre méthode contient les variables suivantes : les paramètres de régularisation λ, α, μ et le nombre de facteurs latents d. La figure 3.6 présente les impacts de ces paramètres sur la performance. Dans chaque graphique, nous traçons l'AUC moyenne (sur tous les pas de temps) lors de la variation de chaque paramètre.

Le nombre de facteurs latents d est le nombre de variables latentes que nous extrayons à partir de données. Comme notre algorithme d'extraction des variables latentes peut être considéré comme un algorithme de réduction de dimension, le nombre de facteurs correspond au nombre réduit de dimensions. Pour obtenir une bonne performance, il faut un certain nombre de variables latentes. Nous rappelons aussi que, la complexité de l'algorithme d'optimisation est en ordre de d^3. Théoriquement, il faut bien choisir d pour avoir un bon compromis entre la performance et le coût de calcul. Dans la figure 3.6a, nous voyons que quand d augmente (de 1), la performance augmente. La performance devient stable quand $d > 40$ (la performance évolue très peu). Cet effet de convergence de d nous suggère qu'un petit nombre de facteurs latents (tel que 40) est suffisant pour obtenir une bonne performance. Notons que, la dimension de données originales (i.e le nombre d'attributs sur les nœuds) est en ordre de centaines de milliers (voir l'annexe A).

Le paramètre λ est un paramètre de régularisation. La figure 3.6b montre que quand λ n'est pas très grand ($\lambda < 10$), il n'a pas de grand impact sur la performance. Quand λ est grand, la performance diminue rapidement. Ce phénomène peut être expliqué comme suit : quand λ est grand, tous les facteurs latents tendent vers zéro à la convergence (voir la fonction objectif 3.5), ils sont moins prédictifs. Il faut donc fixer λ à une valeur faible.

Le paramètre α règle la contribution du graphe social dans le modèle à variable latente (voir la fonction objectif 3.5). Dans la figure 3.6c, nous voyons que quand $\alpha = 0$ (i.e aucune information du graphe social n'est intégrée dans le modèle), la performance est faible. La performance augmente quand on augmente α et atteint son maximum à environ $\alpha = 10$. Une grande valeur de α donne une mauvaise performance de prédiction. Cela suggère que, pour obtenir une bonne performance, il faut une valeur moyenne de α (comprise entre 1 et 100) - cette valeur permet une bonne combinaison des informations de contenus (e.g les attributs sociaux et les variables du SI) et les informations du graphe social.

Le paramètre μ règle la contribution du modèle du pas de temps précédent dans le modèle actuel (dans notre modèle d'apprentissage incrémental - l'équation 3.5). La figure 3.6d présente l'impact de ce paramètre sur la performance de prédiction. La meilleure performance est atteinte lorsque μ est entre 0.1 et 10. Une petite valeur de μ ($\mu \approx 0$) donne une mauvaise performance car dans ce cas, les informations dans le modèle du pas de temps ne sont pas réutilisées. Une grande valeur donne aussi une mauvaise performance. Quand μ

est grand, les facteurs latents des nœuds existants n'évoluent pas d'un pas de temps à l'autre (les positions des nœuds sont contraintes à rester immobiles dans l'espace latent). Dans ce cas, les nouvelles données ne sont pas bien intégrées dans le modèle dans chaque pas de temps, ce qui dégrade la performance de prédiction.

Nous concluons que, pour les trois paramètres λ, α et μ, il faut une configuration raisonnable pour obtenir une bonne performance de prédiction. Il faut également bien choisir le nombre de facteurs latents pour avoir une bonne performance et avec un petit coût de calcul. Dans les applications réelles, nous pouvons trouver la bonne configuration de paramètres en utilisant un jeu de données de validation (les données de même nature que les données sur lesquelles nous souhaitons appliquer la méthode). L'idée est d'évaluer la méthode avec plusieurs configurations de paramètres (sur une grille des valeurs des paramètres). Pour chaque valeur de d, en commençant par $d = 1$, nous choisissons la configuration des paramètres λ, α et μ (à partir d'une grille des valeurs) qui donne la meilleure performance (AUC). Nous faisons augmenter d de façon à trouver la valeur de d à partir de laquelle l'AUC obtenue, en maximisant les autres paramètres sur leur grille, ne varie plus significativement. Notons aussi que le choix de la grille des valeurs des paramètres est empirique et dépend des données et de la capacité de calcul. Dans cette thèse, les choix de grille des valeurs des paramètres ne sont pas les mêmes pour toutes les expérimentations.

3.7.2.3 Tests sur les autres scénarios

Dans cette section nous testons notre méthode et la comparons avec les méthodes de référence dans quelques différentes situations. Dans le premier scénario (section 3.7.2.1), le jeu de données est généré sous l'hypothèse suivante : la variable cible dépend des 3 composants de données explicatives : le graphe social, les attributs sociaux et les variables du SI. Ici, nous voulons tester quelques situations où cette hypothèse n'est pas vérifiée.

Pour les expérimentations dans cette section, nous générons les jeux de données synthétiques avec notre générateur avec la même configuration que dans le premier scénario (section 3.7.2.1), sauf pour la partie concernant la variable cible. Nous appliquons les mêmes méthodes de référence présentées dans les sections précédentes.

Concernant la procédure de réglage des paramètres, nous appliquons la procédure de validation. Dans chaque scénario, nous prenons les données des deux premiers pas de temps ($t = 0, t = 1$) comme données de validation. Nous faisons varier les valeurs des paramètres (λ, α, μ) dans l'ensemble de triplets $\{0.1, 1, 10, 100\} \times \{0, 0.1, 1, 10, 100\} \times \{0, 0.1, 1, 10, 100\}$. Le nombre de facteurs latents d varie dans l'ensemble $\{1, 5, 10, 20, 50\}$. Dans chacun des cas nous appliquons notre méthode sur les données de deux premiers pas de temps ($t = 0, 1$) et calculons la performance de prédiction (AUC) au pas de temps $t = 1$. Rappelons que la performance augmente avec le nombre de facteurs latents d, elle devient stable à partir

d'une certaine valeur de d. Nous avons observé que, dans tous les scénarios, la performance devient stable à partir de $d = 20$ (pour toutes les valeurs des paramètres λ, α, μ). Pour cette raison, nous fixons $d = 20$ dans les tests. Les autres paramètres λ, α, μ sont choisis tels qu'ils donnent la meilleure performance sur les données de validation.

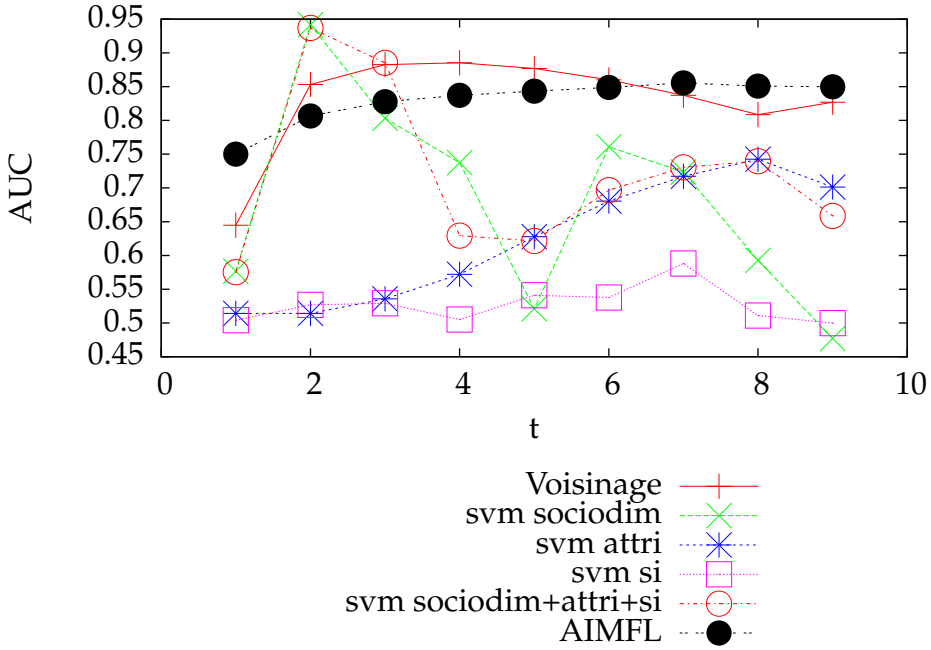


FIGURE 3.7 – Performances (AUC) des différentes méthodes (la variable cible ne dépend que du graphe social)

La variable cible ne dépend que du graphe social Pour générer le jeu de données dans ce scénario, nous réglons les paramètres du générateur de variable cible comme suite : $\alpha_{attri} = \alpha_{si} = 0, \alpha_{soc} = 1.0$. La procédure de réglage des paramètres par validation nous suggérons les valeurs des paramètres $\lambda = 1.0, \alpha = 100, \mu = 0.1$. Nous appliquons notre méthode avec cette configuration et les méthodes de référence. Nous traçons les performances de prédiction (en termes d'AUC) dans la figure 3.7. Comme le graphe social est une source d'information informative, la méthode *voisinage* (basée sur le graphe social) donne une très bonne performance. Une autre méthode basée sur le graphe social, *svm sociodim*, donne des bonnes performances dans quelques pas de temps mais comme expliqué précédemment, elle n'est pas stable. La méthode *svm si*, qui utilise seulement les variables du SI (non-informatives), donne des AUCs autour de 0.5. Concernant la méthode *svm attri* (qui utilise les attributs sur les nœuds sociaux), nous voyons qu'elle peut donner des AUCs bien supérieurs à 0.5. Nous avons réglé $\alpha_{attri} = 0$ mais les attributs portent aussi des informations utiles pour prédire la variable cible. Ce phénomène peut être expliqué comme suit :

dans notre générateur de données synthétiques, les attributs sur les nœuds sont générés de sorte que les attributs sont corrélés avec le graphe social. Plus précisément, les voisins dans le graphe social tendent à avoir les mêmes attributs. Comme les voisins sociaux sont plus susceptibles d'avoir la même valeur de la variable cible, les attributs portent aussi des informations informatives permettant de la prédire.

Dans ce scénario, notre méthode AIMFL donne aussi de bonnes performances de prédiction. Elle est la seconde meilleure méthode parmi toutes les méthodes testées (après la méthode (*voisinage*) et sur les derniers pas de temps elle donne des performances similaires à celle-ci. Notons que l'intérêt de notre méthode est que nous partons d'aucun a priori sur les données, alors que le fait d'utiliser la méthode de voisinage démontre déjà une connaissance sur les données.

La variable cible ne dépend pas du graphe social Dans ce cas, nous avons la configuration de paramètres suivante :$\alpha_{attri} = \alpha_{si} = 1.0, \alpha_{soc} = 0$. La figure 3.8 présente les performances des méthodes à chaque pas de temps dans ce scénario. Comme attendu, les méthodes qui utilisent le graphe social comme (*voisinage* et *svm sociodim*) donnent des mauvaises performances et les méthodes basées sur les attributs (*svm attri*) et les variables du SI (*svm si*) donnent de bonnes performances. La meilleure méthode est *svm sociodim+attri+si* (AUC de 0.8 - 0.9), combinaison de toutes les types de données. Bien que les dimensions sociales ne soient pas informatives, cette méthode a réussi à combiner les deux sources informatives : les attributs sociaux et les variables du SI pour donner un bon modèle prédictif.

Notre méthode est appliquée avec la configuration de paramètre suivante $\lambda = \mu = 1, \alpha = 0$ (en utilisant procédure de réglage des paramètres par validation). Nous voyons que dans ce cas, notre méthode ne peut pas donner de bonnes performances (AUC autour de 0.7). Notre méthode à base de variables latentes est moins bonne que les méthodes basées sur l'apprentissage direct avec les attributs. Dans ce type de situations, les méthodes classiques basées sur les données attribut-valeur sont en effet les plus adaptées.

La variable cible ne dépend que des données de média sociaux La variable cible ne dépend que du graphe social et des attributs sur les nœuds. Pour générer le jeu de données dans ce scénario, nous réglons $\alpha_{attri} = \alpha_{soc} = 1.0, \alpha_{si} = 0$. La figure 3.9 présente les performances des différentes méthodes. Ici, nous voyons que la méthode qui utilise les variables du SI (*svm si*) n'est pas meilleure que le hasard (i.e AUC autour de 0.5). C'est un résultat attendu parce que les variables du SI ne portent pas d'informations utiles pour prédire la variable cible. La méthode *voisinage* donne de bonnes performances en utilisant le graphe social. La méthode *svm sociodim* utilise aussi le graphe social, mais elle n'est pas aussi bonne que la méthode *voisinage*. La meilleure méthode est *svm sociodim+attri+si*, une combinaison

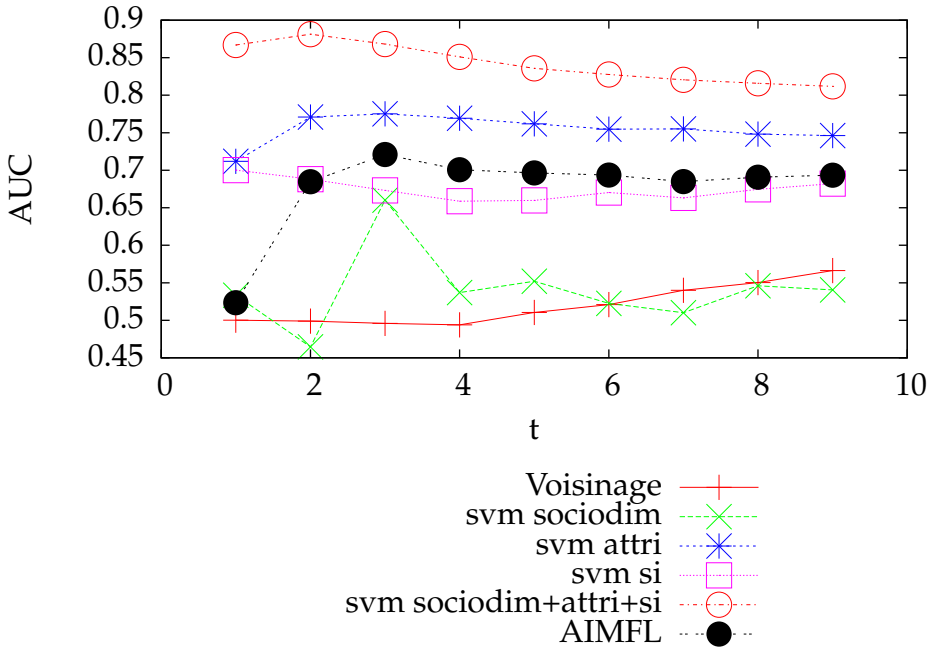


FIGURE 3.8 – Performances (AUC) des différentes méthodes (la variable cible ne dépend pas du graphe social)

de tous les types d'information dans les données (ici, les dimensions sociales et les attributs sont informatifs dans le modèle d'apprentissage de *svm sociodim+attri+si*).

Dans ce scénario, notre méthode AIMFL (nous l'avons appliqué avec $\lambda = 1, \alpha = 10, \mu = 0.1$, ces valeurs sont choisies par le réglage des paramètres par validation) donne des meilleures performances parmi les méthodes testées.

La variable cible ne dépend que des variables du SI Dans ce scénario, la variable cible est indépendante des données issues des média sociaux (le graphe social et les attributs). Le jeu de données est généré avec $\alpha_{attri} = \alpha_{soc} = 0, \alpha_{si} = 1.0$. Les performances de différentes méthodes sont présentées dans la figure 3.10. Comme prévu, le graphe social et les attributs ne portent aucune information informative pour la variable cible (les AUCs de *svm attri*, *voisinage*, *svm sociodim* sont autour de 0.5). La seule source d'information utile pour la prédiction sont les variables du SI ; les méthodes (*svm si* et *svm sociodim+attri+si*) qui utilisent ces variables donnent ainsi de bonnes performances (AUC autour de 0.9).

Notre méthode (appliqué avec les paramètres $\alpha = 0, \mu = \lambda = 1.0$, ces valeurs sont choisies par le réglage des paramètres par validation) peut prédire la variable cible (avec une AUC variée de 0.7 à 0.9). Là encore, notre méthode à base des variables latentes est moins bonne que les méthodes nativement conçues pour les données attribut-valeur.

À partir des scénarios que nous avons examinés, nous remarquons que notre méthode

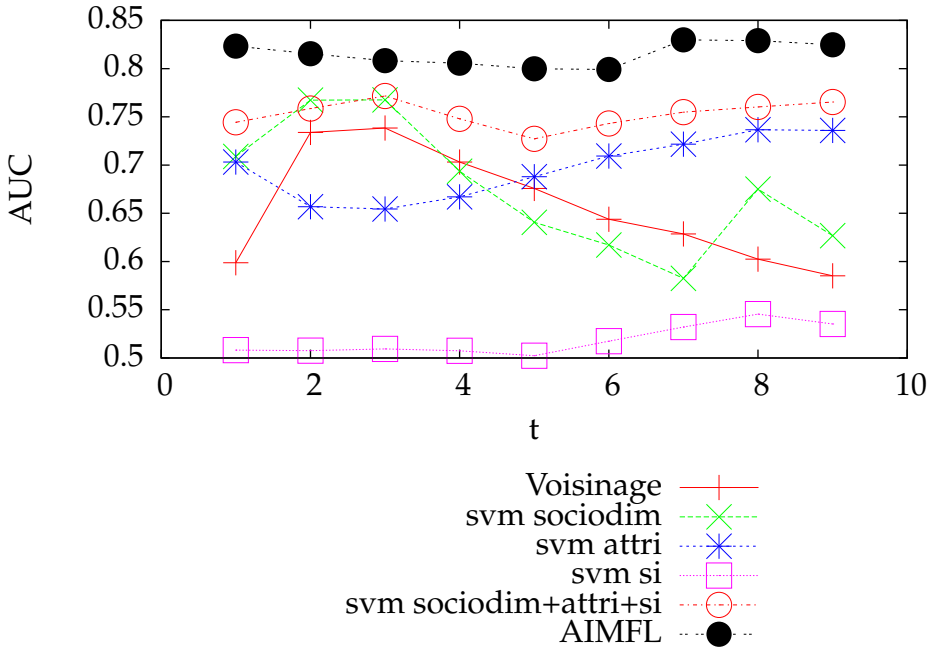


FIGURE 3.9 – Performances (AUC) des différentes méthodes (la variable cible ne dépend que des données de média sociaux)

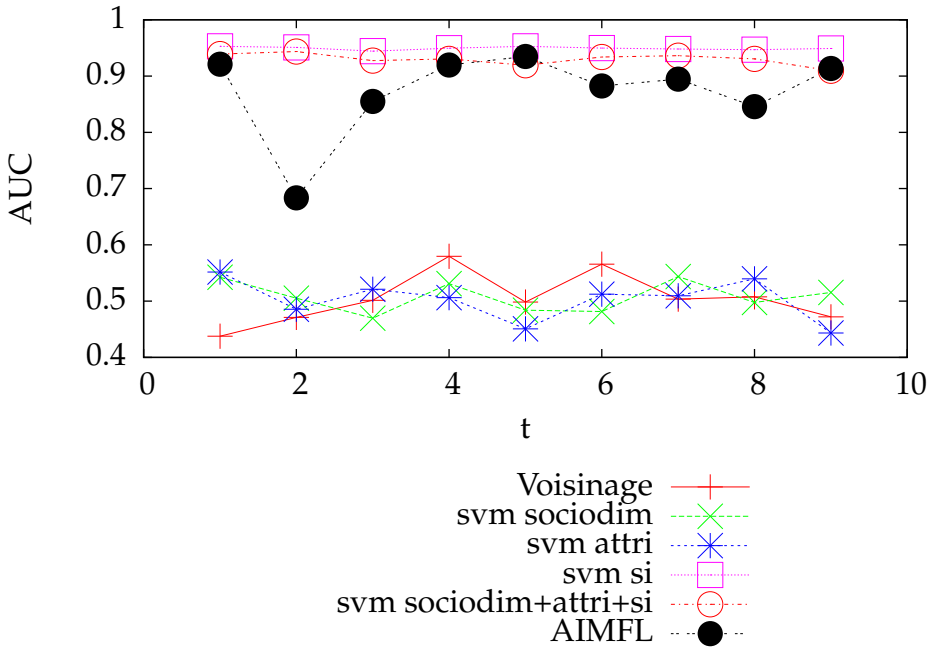


FIGURE 3.10 – Performances (AUC) des différentes méthodes (la variable cible ne dépend que les variables du SI)

marche bien dans les cas où le graphe social est suffisamment informatif. Dans ces cas, notre méthode donne des performances comparables ou meilleures que les meilleures méthodes de référence (les scénario avec $\alpha_{soc} = 1.0$). Par contre, dans les cas où seulement les attributs et/ou les variables du SI sont informatives, notre méthode est moins bonne que la méthode qui utilise directement les attributs et/ou les variables du SI pour l'apprentissage supervisé. Intuitivement, nous pouvons expliquer ce phénomène de manière suivante : dans les cas où le graphe social n'est pas informatif, nous comptons seulement sur les données de type attribut-valeur (les attributs sociaux et les variables du SI) pour prédire le comportement des clients. En utilisant les facteurs latents nous avons perdu une partie de l'information utile dans le processus de transformation des données attribut-valeur en facteurs latents. L'apprentissage supervisé sur les données originales (les attributs) est donc meilleur.

3.7.3 Temps de calcul

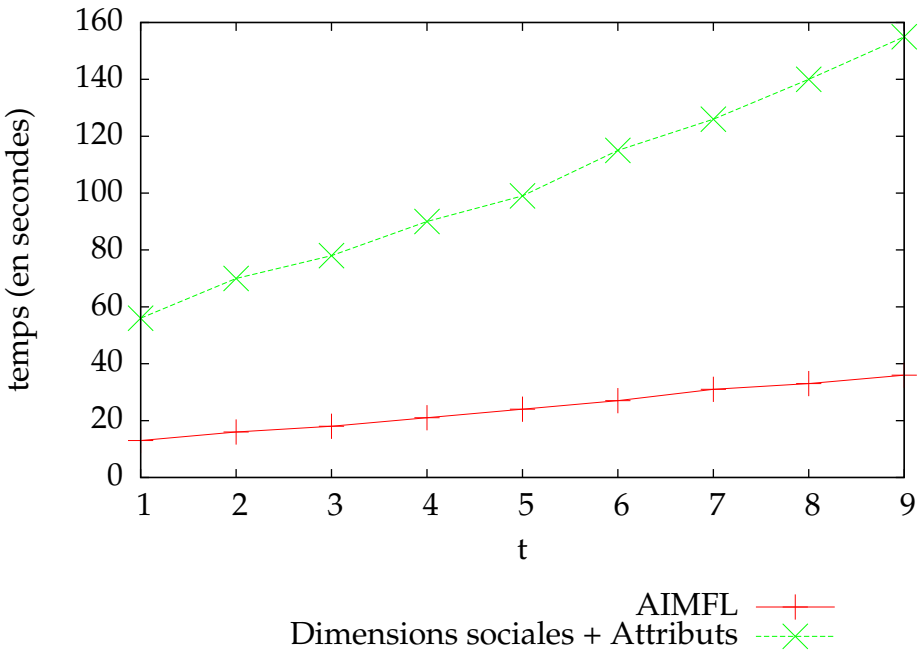


FIGURE 3.11 – Temps de calcul

La figure 3.11 présente le temps de calcul des deux méthodes : notre méthode et la méthode basées sur les dimensions latentes *svm sociodim+attri+si* (les autres méthodes, qui ne prennent en compte qu'une partie de données, sont en général plus rapide que ces deux méthodes). Les méthodes sont implémentées et exécutées sur la même machine (Linux 64 bits, CPU 8x2.1GHz). Pour notre méthode incrémentale, les calculs effectués à chaque pas de temps sont le calcul des facteurs latents (optimisation) et l'apprentissage d'un classifieur

SVM. Pour la méthode *svm sociodim+attri+si*, les calculs sont la classification spectrale du graphe social et l'apprentissage d'un classifieur SVM. La figure 3.11 indique un gain significatif en termes de calcul en utilisant l'apprentissage incrémental avec notre méthode par rapport à la méthode *svm sociodim+attri+si*. Au pas de temps t, notre méthode prend seulement le RSA courant $\mathcal{G}^{agg}(t)$ pour mettre à jour les facteurs latents, alors que la méthode *svm sociodim+attri+si* doit considérer toutes les données au passé $\mathcal{G}^{agg}(t)$ pour apprendre son modèle. La figure 3.11 est une illustration de cette analyse de complexité : l'apprentissage avec des données agrégées ($\mathcal{G}^{agg}(t)$) devient de plus en plus coûteux alors que l'apprentissage incrémental exige seulement le temps de traiter les données courantes ($\mathcal{G}(t)$).

3.8 Conclusion

Dans ce chapitre, nous avons introduit une nouvelle méthode d'apprentissage inspirée des modèles à facteurs latents. Cette méthode peut être considérée comme un algorithme d'apprentissage de représentation (et réduction de dimension). Elle permet de représenter des individus dans un espace latent à dimension faible. La méthode d'apprentissage proposée est applicable sur les données sous forme de *réseau social attribué*. Notre méthode (AIMFL), basée sur la factorisation de matrice, permet d'utiliser conjointement les informations de type de contenus et les informations sociales (relationnelles mais aussi textuelles ou tout autre type d'attributs sociaux encodés dans les RSAs) pour extraire des variables latentes.

Nous avons aussi présenté comment de la méthode dans le cadre d'une stratégie de CRM intercanale. L'idée est de représenter les données intercanales sous forme de graphes sociaux attribués, d'extraire les variables latentes et d'utiliser les variables latentes comme variables explicatives pour le problème de prédiction des actes commerciaux. Cette méthode permet aussi un apprentissage incrémental dans le sens où les variables latentes sont mises à jour avec des données courantes à chaque pas de temps. Ainsi, ceci permet de traiter des lots de nouvelles données sociales potentiellement volumineuses, fractionnées en incréments de moindre taille.

Dans la partie d'expérimentation, nous avons simulé les données issues de deux canaux : les média sociaux et le SI client. Les données issues de média sociaux sont sous forme d'un graphe social attribué - les nœuds sociaux représentent les clients et les attributs représentent les contenus créés par les clients dans les média sociaux. Les données du SI sont des variables explicatives sur chaque nœud. En utilisant un jeu de données synthétiques, nous avons montré que notre algorithme (AIMFL) peut extraire des variables informatives pour prédire une variable cible qui dépend à la fois du graphe social, des attributs sociaux du graphe mais aussi des variables du SI. Notre méthode donne des performances compa-

rables à la meilleure méthode de référence, qui est basée sur l'apprentissage supervisé et qui utilise conjointement toutes les sources d'informations informatives.

Nous avons aussi examiné plusieurs scénarios, dans lesquels seulement une partie de données sont informatives pour la variable cible. Nous remarquons que notre méthode est meilleure ou comparable avec les méthodes de référence (apprentissage supervisé en utilisant conjointement le graphe social, les attributs et les variables du SI) dans les cas où le graphe social est suffisamment informatif. Dans les cas contraires, c'est-à-dire les cas où le graphe social n'a pas un caractère d'homophilie assez fort, notre méthode est moins bonne. A ce stade, nous ne pouvons pas dire si l'hypothèse d'homophilie de notre méthode est gênante. Notre objectif est maintenant de tester sur des données réelles. Si leur caractère d'homophilie n'est pas démontré par faute de temps, nous verrons néanmoins que notre méthode présente de bons pouvoirs prédictifs, ce qui nous permet d'affirmer que cette hypothèse n'est en tous les cas pas bloquante pour notre contexte applicatif.

Une limitation de notre méthode est qu'il y a un certain nombre de paramètres à régler. La procédure de réglage des paramètres a besoin d'un jeu de données de validation et cela demande du temps.

Pour conclure, notre méthode est une méthode d'apprentissage de représentation sur des données complexes (encodées sur les réseaux sociaux attribués). L'idée est de transformer les données en une représentation à dimension faible (les variables latentes). Cette méthode est capable de traiter les données issues de plusieurs sources et de différentes formes : le graphe d'interactions, les contenus (textes) et les données tabulaires. Malgré quelques limitations (concernant la procédure de réglage des paramètres, les meilleures performances ne sont pas garanties dans tous les cas), la méthode proposée a quelques avantages correspondant à certaines problématiques que nous avons identifiées pour le contexte du CRM intercanale (cf. Chapitre 1). Un avantage est la capacité de notre méthode à apprendre de manière incrémentale, dans le sens où les facteurs latents sont mis à jour à chaque pas de temps avec les données courantes. Notre méthode est adaptée à la dynamique : elle prend en compte des nouveaux types de contenus (en créant les nouveaux nœuds d'attribut). Un autre point fort de la méthode est la capacité à passer à grande échelle ; l'algorithme d'apprentissage que nous avons proposé est parallélisable. Dans la suite de cette thèse, nous mettons en œuvre la méthode sur des jeux de données réelles (données de CRM intercanal, mais aussi données collectées sur Twitter) pour évaluer sa performance en situation réelle.

CHAPITRE 4

APPLICATIONS DE NOTRE MÉTHODE POUR DIFFÉRENTS PROBLÈMES DE PRÉDICTION

Sommaire

Dans ce chapitre nous présentons les applications de notre méthode (AIMFL) sur les 2 jeux de données. Avec le premier jeu de données, recueillis de Twitter, nous essayons de prédire des activités des utilisateurs sur Twitter vis-à-vis de la marque : qui parlera de la marque. Le deuxième jeu de données est un jeu de données intercanales - nous avons réussi à constituer ce jeu de données en faisant la jointure entre un forum d'entraide de la marque et les données client (le SI client). Avec le deuxième jeu de données, nous testons notre méthode pour le problème de prédiction des actes commerciaux des clients.

4.1 Prédire qui parlera de la marque sur Twitter

Dans cette section, nous nous concentrons sur le problème de la prévision des activités des utilisateurs dans les médias sociaux. Notre défi consiste à considérer les événements réels tels que les messages publiés ou la transmission de ceux reçus aux amis, la connexion à de nouveaux amis, et de fournir de prévision en temps réel (ou quasi réel) des nouveaux événements. L'événement que nous essayons de prévoir est le fait qu'un utilisateur parle de la marque dans ses *tweets*. Nous utilisons un jeu de données collecté à partir de Twitter. Nous décrivons d'abord le processus de collecte de données et les pré-traitements pour rendre les données au format du RSA. [1]

4.1.1 Description du jeu de données

Nous avons collecté le jeu de données via Twitter API [2]. Ces données sont datées dans la période du juillet jusqu'au début décembre 2012 et elles concernent les *followers* de Sosh [3] sur Twitter (c'est-à-dire les internautes qui se sont abonnés à @Sosh_fr sur Twitter). Pour collecter les données, nous avons créé un *crawler* et nous l'avons laissé tourner pendant la période du juillet jusqu'au début décembre 2012. Le schéma 4.1 illustre le fonctionnement de notre *crawler*. Notre *crawler* est capable, pour chaque *follower*, de collecter toutes ses *tweets*, sa liste de *followers* et sa liste de *retweets* (les identifiants des *tweets* sur lesquels il a cliqué « *retweet* »). La collection des données a été faite en continu, en parcourant tous les utilisateurs en *round robin* : le *crawler* récupère les données pour le premier utilisateur dans la liste, puis le second et ainsi de suite jusqu'au dernier, puis un tour est recommencé avec le premier utilisateur, etc. Pendant la période de juillet jusqu'au début décembre 2012, de nouveaux *followers* de @Sosh_fr se sont inscrits et notre *crawler* a aussi ajouté ces nouveaux *followers* au fil du temps.

1. Les résultats principaux présentés ici ont été publiés dans Discovery Science 2014 [LTBCK14].
2. https ://dev.twitter.com/docs/api
3. Sosh est une marque française de téléphonie mobile, sans engagement, développée en France par l'opérateur Orange depuis le 6 octobre 2011

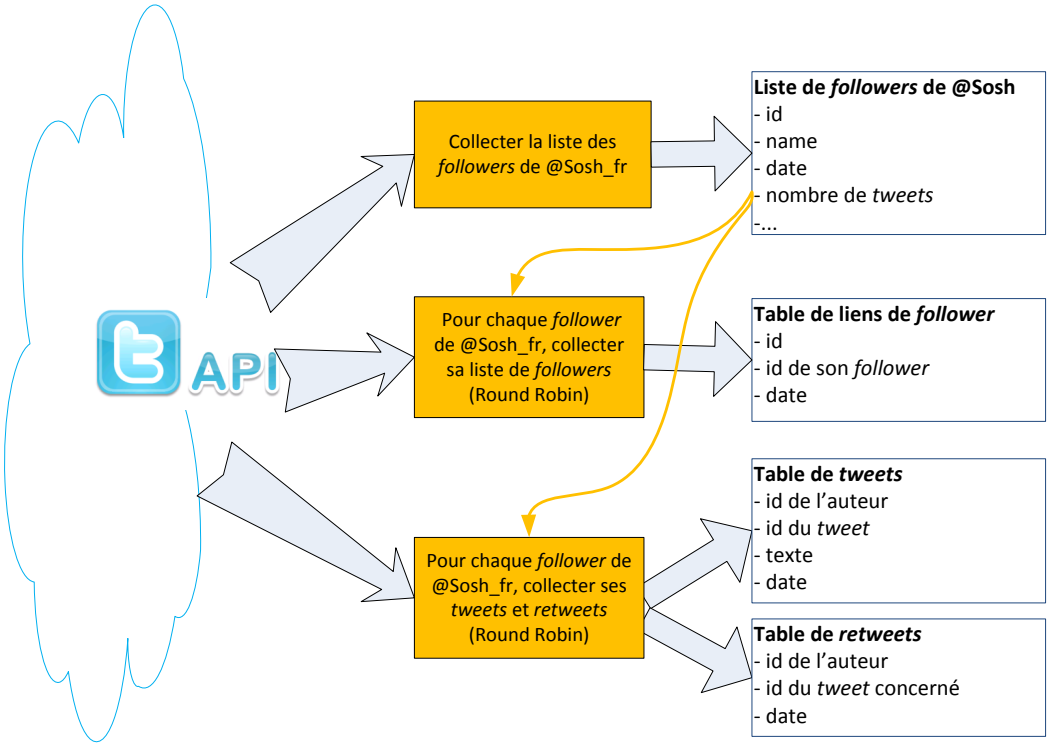


FIGURE 4.1 – Processus du crawl de données via *Twitter API*. Date du crawl : 07/2012 - 12/2012

De cette manière, nous avons recueilli les données assez régulièrement pour être en mesure de construire 21 sous-jeux de données instantanés échantillonnés sur la base d'une semaine. La première semaine commence le 15/07/2012. Dans cette section, les semaines sont numérotées de $t = 0$ jusqu'à $t = 20$. La taille des données dans chacune des 21 semaines est présentée dans la Table B.1 de l'annexe B.

4.1.2 Construction des RSAs

Nous considérons les liens avec les *followers* et les *retweets* comme les informations relationnelles entre les internautes et les contenus des *tweets* comme les attributs décrivant les internautes. En nous basant sur cette idée, nous construisons, pour chaque semaine t, un RSA $\mathcal{G}(t)$ de la manière suivante :

Le graphe social Nous établissions un lien social entre deux individus s'ils sont connectés par un lien de *follower* (l'un suit l'autre) ou s'ils ont *retweeté* un *tweet* commun dans la semaine t. Nous pouvons voir que c'est une somme de deux graphes : le graphe de *follower* et le graphe de *co-retweet*.

Nous supposons que le graphe social est fondé sur le caractère d'*homophilie*. Notre hypothèse est en effet que si deux individus connectés par un lien de *follower* (l'un suit l'autre) ou s'ils ont *retweeté* les mêmes *tweets*, ils pourraient avoir des comportements identiques, tous les cas, pour notre expérimentation, ils émettent un *tweet* parlant du même sujet, ici la marque Sosh.

Le graphe d'attribut Le graphe d'attribut modélisant les profils thématiques des individus est obtenu à partir du contenu de leurs *tweets*. Nous considérons chaque mot dans le(s) *tweet(s)* des utilisateurs comme un attribut. Nous mettons un lien d'attribut entre un utilisateur et un mot si l'utilisateur a posté un *tweet* contenant le mot dans la semaine t. Le lien est pondéré par le nombre de fois que cela s'est produit. Nous avons utilisé Patatext [1] pour traiter les textes (*tokenization*) et construire le graphe d'attribut. Dans le pré-traitement, nous avons enlevé quelques mots vides [2] (qui ne portent pas d'informations utiles) dans la liste de mots.

La taille de chacun des RSAs $\mathcal{G}(t)$, $t = 0, 1, ..., 19$ est présentée dans la Table B.2 de l'annexe B.

4.1.3 Notre problème de prédiction

Nous nous intéressons à prédire qui va parler de la marque Sosh dans la semaine $t + 1$ à partir de données jusqu'à la semaine t incluse. Une personne parle de Sosh sur Twitter si elle

1. Patatext est un outil de traitement de texte développé à Orange Labs
2. Nous avons utilisé la liste suggérée à l'adresse suivante http ://www.ranks.nl/stopwords/french

poste un *tweet* contenant le mot « Sosh » ou bien mentionne @Sosh_fr dans un *tweet*. Parmi les *followers* de @Sosh_fr, ceux qui parlent de « Sosh » sont souvent les clients de Sosh ou tout simplement les internautes intéressés par la marque qui pourraient être les futurs clients de la marque. Voici quelques exemples des « tweets » qui parlent de Sosh :

- « JE pense que je vais passer sur SOSH moi ! »
- « La nouvelle pub de sosh est kiffante ! »
- « sosh depuis avril je suis chez sosh je n'aie jamais eu la 3g je vais retourner chez sfr s'il ne regle pas le problème »
- « Promotions accessoires mobile et internet sur @Sosh_fr http ://t.co/E7aYdSae »

En termes de Social CRM, le fait que quelqu'un parle de la marque exprime un certain niveau d'engagement entre cet individu et la marque (cf. section 1.2.2). Cet événement est donc intéressant à prédire.

À la fin de la semaine t, nous considérons le problème de la prédiction comme un problème de classification où les individus portant une étiquette positive correspondent à ceux qui parleront de Sosh dans la semaine suivante. La Figure 4.2 présente le nombre d'individus et le nombre d'étiquettes positives dans chaque semaine. La proportion des individus portant une étiquette positive dans chaque semaine est relativement faible (environ 1%).

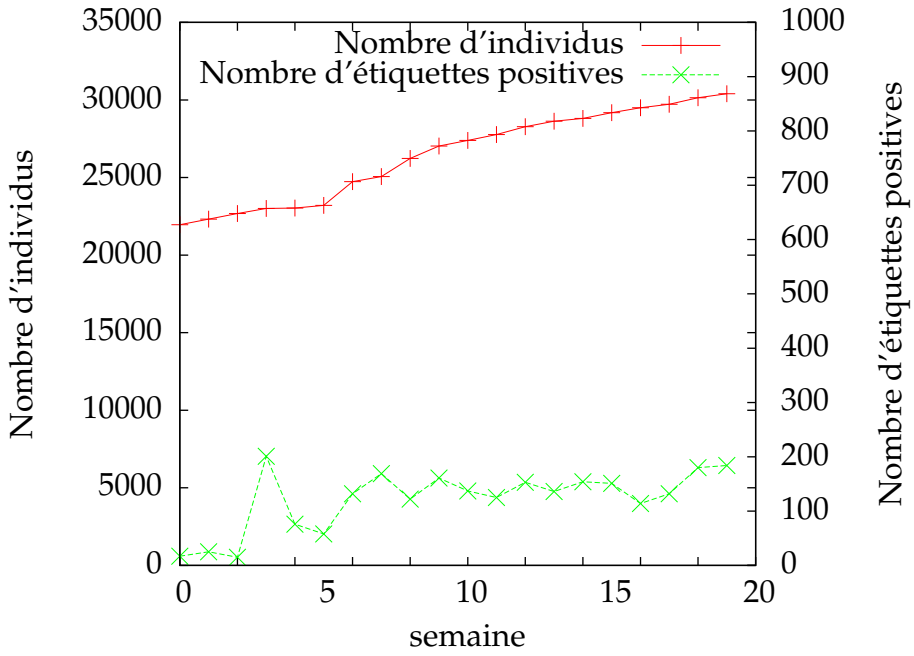


FIGURE 4.2 – Nombre d'individus et nombre des étiquettes positives à chaque semaine

4.1.4 Prédiction avec notre méthode

Nous appliquons notre méthode (AIMFL) pour apprendre des facteurs latents des individus à chaque pas de temps (i.e à la fin de chaque semaine). Ces facteurs latents sont ensuite utilisés comme variables explicatives pour le problème de classification à chaque pas de temps que nous avons définis au-dessus. Plus précisément, à la fin de la semaine t, nous utilisons des facteurs latents des individus pour apprendre un classifieur (SVM), les étiquettes positives correspondent aux individus qui parlent de Sosh dans la semaine $t+1$. À la fin de la semaine $t+1$, nous utilisons ainsi le classifieur appris précédemment et les facteurs latents actuels pour prédire qui parleront de Sosh dans la semaine $t+2$.

Nous utilisons l'aire sous la courbe ROC (AUC) [Bra97] pour mesurer la performance de prédiction. Nous rappelons que cette mesure est une bonne mesure de performance pour comparer différentes méthodes, même si la répartition des exemples positifs et négatifs dans nos données est très inégale.

Nous utilisons l'approche FRRM pour apprendre les facteurs latents (cf. équation 3.5 et nous utilisons le terme de régularisation normalisé). Nous rappelons que l'approche FRRM permet d'exploiter le caractère d'homophilie dans les données relationnelles. De même nous avons fixé a priori le nombre d'itérations dans notre algorithme d'optimisation (ALS) à 20 (Et lorsque que nous avons testé plusieurs valeurs, nous n'avons observé aucune amélioration de la performance au-delà de 20 itérations).

Les paramètres ont été choisis en utilisant des données de validation. Nous utilisons les données des deux premiers pas de temps comme données de validation. Le nombre de facteurs latents d est fixé a priori à 10. Nous avons observé que d n'a pas d'impact significatif sur la performance. Nous étudierons de manière approfondie l'impact de ce paramètre par la suite. La procédure de réglage des paramètres (λ, α, μ) est la suivante : nous faisons varier les valeurs des paramètres (λ, α, μ) dans l'ensemble de triplets $\{1, 10, 50, 100\} \times \{1, 10, 50, 100\} \times \{1, 10, 50, 100\}$. Dans chacun des cas nous appliquons notre méthode sur les données des deux premiers pas de temps ($t = 0, 1$) et nous calculons la performance de prédiction au pas de temps $t = 1$. Nous choisissons ensuite le triplet (λ, α, μ) qui donne la meilleure performance. Suite à cette étape, nous avons fixé les paramètres comme suit : $\lambda = 50, \alpha = 100, \mu = 100$.

4.1.5 Les méthodes de référence

À chaque pas de temps (de la semaine) t nous appliquons les techniques suivantes pour comparer avec notre méthode :

Méthode triviale 1 Puisqu'un internaute peut parler de Sosh plus d'une fois dans la période d'observation, il est intéressant de savoir s'il s'agit d'une action répétée : si quelqu'un

a parlé de Sosh, il est probable qu'il en parlera à nouveau. Nous construisons une méthode triviale basée sur cette observation : au pas de temps t, nous mettons une étiquette positive sur tous ceux qui portaient une étiquette positive (au moins une fois) dans le passé (avant t).

Méthode triviale 2 Sur Twitter, il y a des utilisateurs très actifs qui écrivent beaucoup de *tweets*, ont beaucoup de *followers*. Naturellement, ces utilisateurs sont plus enclins à parler de Sosh. À partir de cette observation, nous créons une méthode triviale pour prédire qui parlera de Sosh basée sur le niveau d'activité des internautes sur Twitter. Nous considérons le nombre de *tweets* postés comme une mesure d'activité des internautes sur Twitter. À chaque semaine t, nous comptons le nombre de *tweets* postés jusqu'à t pour chacun des individus, nous considérons ce nombre comme un score de prédiction pour prédire qui parleront de Sosh la semaine prochaine $t + 1$.

Méthode de voisinage Il s'agit de modéliser l'influence dans le graphe social (voir 2.2). À chaque pas de temps t, nous construisons un graphe social qui contient tous les liens sociaux datés jusqu'à t (les liens de *follower* et les liens de *co-retweets*. Ensuite, nous calculons le score de prédiction pour chaque individu dans ce graphe ainsi : au pas de temps t, le score d'un individu est égal à la proportion de ses voisins dans le graphe qui ont parlé de Sosh dans passé (avant t).

Dimensions sociales Nous avons décrit en détail cette méthode dans la section 3.7.2.1. Cette méthode est une adaptation de la méthode proposée dans [TL11]. Elle utilise aussi le graphe social. À chaque pas de temps, nous extrayons les *dimensions sociales* des individus dans ce graphe en utilisant le clustering spectral. Ces dimensions sociales sont ensuite utilisées comme variables explicatives pour entraîner un classifieur SVM, les étiquettes positives correspondent aux internautes qui ont parlé de Sosh à $t + 1$. Il s'agit de même procédure que notre méthode, mais ici les facteurs latents sont remplacés par les dimensions sociales. Nous avons fixé le nombre de dimensions sociales à 10 comme nous n'avons observé aucune amélioration de performance en augmentant le nombre de dimensions sociales au-delà de 10.

SVM avec les attributs Nous utilisons la classification supervisée SVM sur les attributs, i.e. les mots dans les *tweets* des individus. À chaque pas de temps t, nous formons un classifieur SVM avec les étiquettes positives aux internautes qui ont parlé de Sosh à $t + 1$. Les variables explicatives sur un individu sont les fréquences des mots dans ses *tweets* datés avant ou dans la semaine t.

Dimensions sociales + attributs Il s'agit de combiner les deux méthodes précédentes. Cette méthode utilise la classification supervisée (SVM), les variables explicatives sont les dimensions sociales et les fréquences des mots dans les *tweets*. Cette combinaison permet d'exploiter simultanément le graphe social et les attributs.

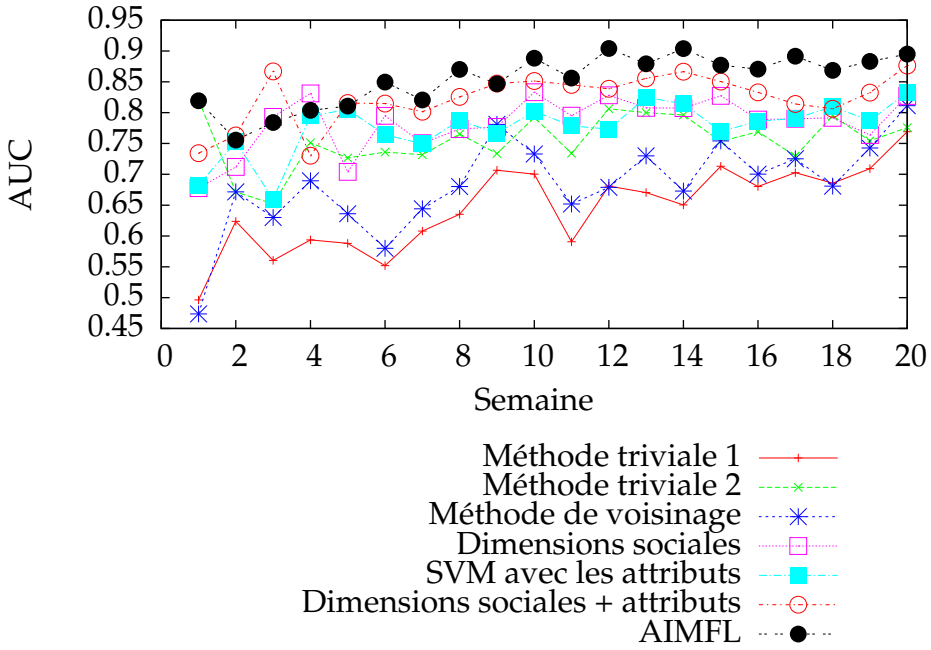


FIGURE 4.3 – Performances (AUC) des différentes méthodes

Nous remarquons que, parmi les méthodes listées au-dessus, seule la dernière utilise à la fois le graphe social et les attributs (les mots sur les *tweets*).

Nous remarquons aussi que toutes les méthodes de référence utilisent les données disponibles dans le passé pour construire leur modèle prédictif à un instant donné. Dans les expérimentations, nous avons essayé d'utiliser les données dans une fenêtre temporelle de différentes tailles pour appliquer les méthodes de référence. Nous avons essayé les fenêtres de taille de 1 semaine, de 4 semaines et de taille infinie (toutes les données dans le passé sont utilisées). Nous avons observé que, en général, pour toutes les méthodes de référence listées au-dessus, la meilleure performance est obtenue lorsque toutes les données dans le passé sont utilisées. Pour cette raison, ici nous ne présentons que les performances des méthodes de référence avec la fenêtre de taille infinie, c'est-à-dire le cas où toutes les données disponibles dans le passé sont utilisées.

4.1.6 Performance

La figure 4.3 présente les performances des différentes méthodes de prédiction, y compris notre méthode, pour les 20 pas de temps $t = 1, 2, ..., 20$. Tout d'abord, nous remarquons que les deux types d'information dans Twitter : les informations relationnelles (c'est-à-dire le graphe social) et les contenus (c'est-à-dire les mots dans les *tweets*) sont informatives. À part la méthode basée sur le voisinage, les méthodes qui utilisent le graphe social ou les

mots dans les *tweets* sont meilleures en AUC que les méthodes triviales qui n'utilisent pas ces données. La méthode basée sur le voisinage n'est pas bien adaptée : elle donne une performance inférieure à celle de la deuxième méthode triviale.

Les méthodes qui utilisent à la fois le graphe social et les mots ("Dimensions sociales + attributs" et notre méthode AIMFL) sont les meilleures. Autrement dit, on peut améliorer la performance de prédiction en exploitant simultanément les informations relationnelles et les contenus.

Sauf quelques perturbations aux premiers pas de temps, notre méthode incrémentale AIMFL donne les meilleures performances. Nous en concluons que, en exploitant à la fois les informations relationnelles et les contenus, on peut améliorer sensiblement la performance de prédiction. De plus, sur cette expérience, le caractère incrémental et le fait que l'apprentissage n'utilise pas l'ensemble de l'historique des données n'est pas un handicap. Notre méthode peut atteindre des performances comparables ou meilleures que des techniques non-incrémentales, en particulier ici la méthode de l'état de l'art qui nous sert de référence.

Un avantage de notre technique incrémentale par rapport à la méthode ("Dimensions sociales + attributs") est qu'elle prend en considération seulement les données courantes ($\mathcal{G}(t)$) à chaque pas de temps t. Cela permet de réduire significativement le temps de calcul. La figure 4.4 montre les temps de calcul de notre méthode incrémentale (AIMFL) et la meilleure méthode non-incrémentale ("Dimensions sociales + attributs") à chaque pas de temps. Les deux méthodes sont implémentées et exécutées sur la même machine (Linux 64 bits, CPU 8x2.1GHz). La figure 4.4 confirme empiriquement que nous pouvons gagner en temps de calcul en utilisant l'apprentissage incrémental avec notre méthode AIMFL.

4.1.7 Effet des paramètres de la méthode AIMFL

Nous examinons la sensibilité des paramètres importants de notre méthode incrémentale : λ, α, μ et le nombre de facteurs latents d. Pour chaque configuration des paramètres, nous calculons la moyenne des performances dans tous les pas de temps. Pour les trois paramètres λ, α, μ, nous observons les impacts similaires que ceux décrits dans la section 3.7.2.2 (le premier test avec les données synthétiques). Comme montré dans la figure 4.5a, λ devrait être supérieur à 1 et ne devrait pas être très grand (>100) pour maintenir la performance. Concernant α (figure 4.5b), de trop petites ou de trop grandes valeurs du paramètre α donnent une faible performance. Une grande valeur de α correspond à un modèle où les interactions sociales jouent un rôle important. Quand $\alpha = 0$, aucune interaction sociale n'est utilisée. La valeur maximale d'AUC est atteinte quand $\alpha = 100$. L'effet du paramètre μ est présenté dans la figure 4.5c. Nous rappelons que ce paramètre contrôle la contribution des facteurs latents appris au pas de temps précédent. Nous voyons que quand μ augmente, la

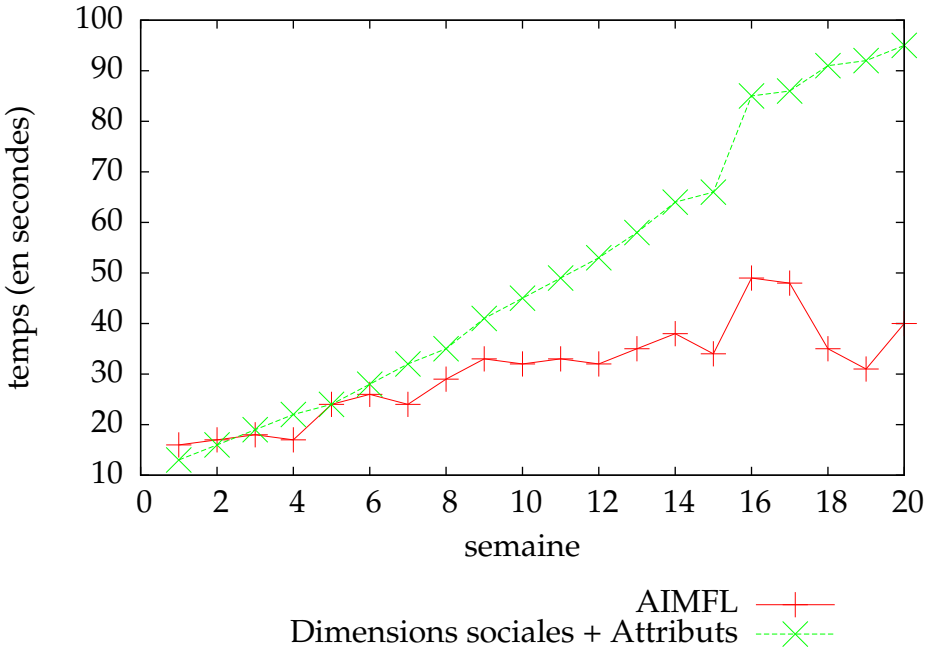


FIGURE 4.4 – Temps de calcul

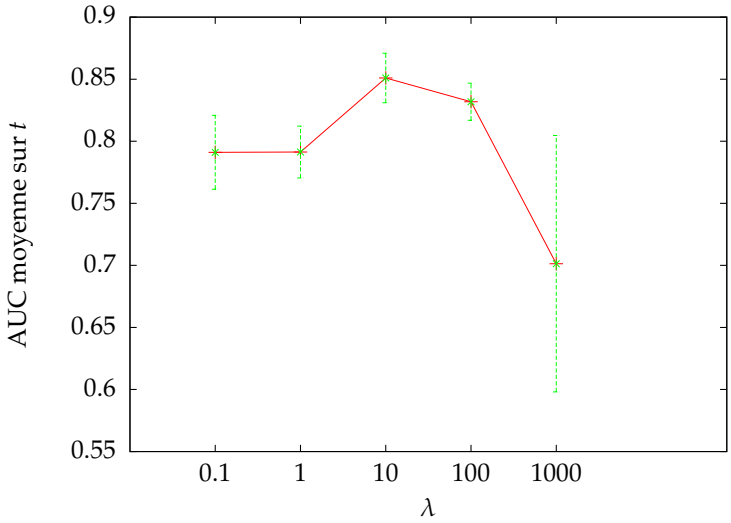
performance augmente et elle atteint son maximum avec $\mu = 100$.

Concernant le nombre de facteurs latents d, nous n'avons pas observé une influence significative de ce paramètre sur la performance que d vaille 1 ou plus. Nous nous sommes aperçus qu'il n'y avait qu'une seule dimension latente informative pour la prédiction de tweet parlant de Sosh. En utilisant *l'analyse en composantes principales*, nous avons détecté que la première composante principale représente sans surprise 95% de la variance. C'est-à-dire qu'il n'y a quasiment qu'une seule composante indépendante dans les d facteurs latents. Dans le paragraphe suivant nous cherchons à comprendre à quoi correspond cette dimension latente unique.

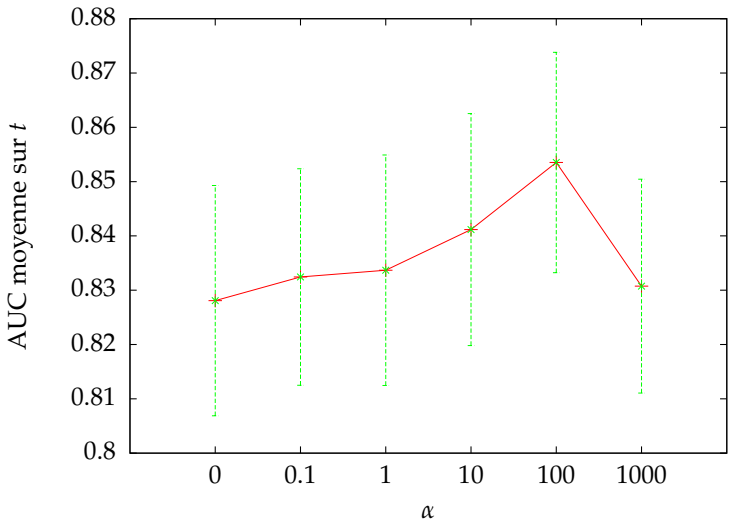
4.1.8 À quoi correspondent les dimensions latentes ?

Comme il n'y a qu'une seule dimension latente informative parmi les d facteurs latents, une question importante se pose : qu'est-ce qui correspond à cette dimension latente ? De plus, pourquoi cette dimension latente est-elle informative pour la variable cible que nous essayons de prédire ? Pour répondre à ces questions, nous avons construit un ensemble de variables explicatives à la main et nous avons examiné la corrélation entre ces variables explicatives et notre facteur latent.

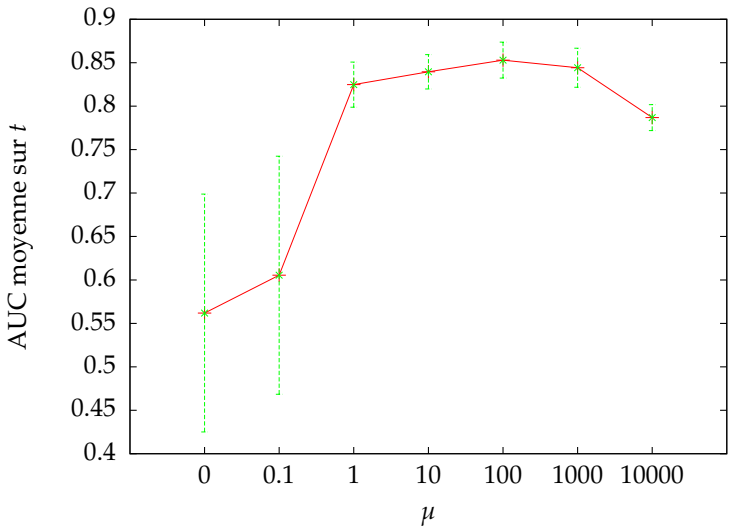
Nous avons choisi le pas de temps $t = 12$ et construit des variables explicatives à ce



(a) α ($d = 10$, $\alpha = 100$ et $\mu = 100$)



(b) α ($d = 10$, $\lambda = 50$ et $\mu = 100$)



(c) μ ($d = 10$, $\lambda = 50$ et $\alpha = 100$)

FIGURE 4.5 – Effet des paramètres de la méthode AIMFL

pas de temps [1]. Nous avons construit un RSA $\mathcal{G}^T$ à partir des données dans une fenêtre de temps T avant la semaine $t = 12$ et construit à la main les variables explicatives que nous pouvons imaginer (à partir des voisins sociaux, des étiquettes-« qui a parlé de Sosh »et des attributs (des mots)) sur chaque individu dans ce RSA. La liste des variables que nous avons calculées, pour une fenêtre T et pour chaque individu est décrite dans le paragraphe suivant.

Construction des variables explicatives pour expliquer la dimension latente Après avoir construit un RSA $\mathcal{G}^T$ à partir des données dans la fenêtre temporelle T, nous calculons les variables explicatives à partir de ce RSA. Les variables que nous allons calculer concernent les éléments suivants : certaines mesures de centralité (nombre de voisins, dégrée) dans le graphe social, le nombre de liens d'attribut, les étiquettes, proportions des étiquettes positives dans le voisinage, etc. Voici les notations utilisées dans ce paragraphe :

- $\mathcal{N}_s^T(i)$: l'ensemble de voisins de l'individu i dans le graphe social.
- $\mathbf{S}^T$: la matrice d'adjacence du graphe social ($\mathbf{S}_{ij}^T$ est le poids du lien (i, j)).
- $\mathcal{N}_a^T(i)$ est l'ensemble de voisins d'attribut de i dans le graphe d'attribut.
- $\mathbf{A}^T$ est la matrice d'attribut. Dans nos données, $\mathbf{A}_{ik}^T$ est la fréquence du mot k dans les *tweets* de i.

Voici la liste des variables que nous avons calculées, pour une fenêtre T et pour chaque individu i :

1. $label^T$

   C'est une variable binaire qui indique si l'individu i parle de Sosh dans la fenêtre temporelle T. Dans le graphe d'attribut, cette variable indique s'il existe un lien d'attribut entre i et le mot « sosh »ou « @Sosh_fr ».

2. $count\_label^T$

   C'est le nombre de fois que l'individu i parle de Sosh dans la fenêtre temporelle T. Dans le graphe d'attribut, cette variable est le nombre de liens d'attribut entre i et le mot « sosh »ou « @Sosh_fr ».

3. $count\_neighbor^T$

   C'est le nombre des voisins sociaux dans le RSA. Nous avons

   $$count\_neighbor^T(i) = |\mathcal{N}_s^T(i)|$$

   On voit que $count\_neighbor^T$ est le nombre des individus avec lesquels un individus i a établi une relation ou interaction (de *follower* ou de *co-retweet*) dans la fenêtre temporelle.

1. Nous avons répété cette expérimentation sur autres pas de temps et obtenu des résultats similaires.

4. soc_degree^T

   C'est le degré de l'individu dans le graphe social. Parce que notre graphe social est pondéré, soc_degree^T est différent de $count\_neighbor_T$:

   $$soc\_degree^T(i) = \sum_{j \in \mathcal{N}_s^T(i)} \mathbf{S}_{ij}^T$$

   On voit que $soc\_degree^T(i)$ est le nombre de relations (de *follower* ou de *co-retweet*) de l'individu i dans la fenêtre temporelle T.

5. $count\_attri^T$

   C'est le nombre d'attributs auxquels l'individu est connecté par un lien d'attribut. Nous avons

   $$count\_attri^T(i) = |\mathcal{N}_a^T(i)|$$

   Autrement dit, $count\_attri^T(i)$ est le nombre de mots uniques que l'individu i a utilisé dans ses *tweets* dans la fenêtre T.

6. $attri\_degree^T$

   C'est le degré d'attribut de l'individu dans le graphe d'attribut :

   $$attri\_degree^T(i) = \sum_{k \in \mathcal{N}_a^T(i)} \mathbf{A}_{ik}^T$$

   $attri\_degree^T(i)$ est donc le nombre total de mots dans les *tweets* dans la fenêtre T.

7. $mean\_attri\_degree^T$

   C'est le degré moyen de l'individu dans le graphe d'attribut :

   $$mean\_attri\_degree^T(i) = \frac{\sum_{k \in \mathcal{N}_a^T(i)} \mathbf{A}_{ik}^T}{|\mathcal{N}_a^T(i)|}$$

   Il s'agit de la moyenne des fréquences de tous les mots dans les *tweets* de l'individu i dans T

8. $label\_neighbor^T$

   C'est le nombre des voisins sociaux de i qui ont parlé de Sosh dans T :

   $$label\_neighbor^T(i) = \sum_{j \in \mathcal{N}_s^T(i)} label^T(j)$$

9. $count\_label\_neighbor^T$

   C'est la somme de $count\_label^T$ sur tous les voisins de i :

$$count\_label\_neighbor^T(i) = \sum_{j \in \mathcal{N}_s^T(i)} count\_label^T(j)$$

Il s'agit le nombre de fois qu'un voisin social de i a parlé de Sosh dans T

10. $weighted\_label\_neighbor^T$

    C'est la somme de $label^T$ sur tous les voisins de i, les termes sont pondérés par les poids des liens sociaux :

$$weighted\_label\_neighbor^T(i) = \sum_{j \in \mathcal{N}_s^T(i)} \mathbf{S}_{ij}^T label^T(j)$$

11. $weighted\_count\_label\_neighbor^T$

    C'est la somme de $count\_label^T$ sur tous les voisins de i, les termes sont pondérés par les poids des liens sociaux :

$$weighted\_count\_label\_neighbor^T(i) = \sum_{j \in \mathcal{N}_s^T(i)} \mathbf{S}_{ij}^T count\_label^T(j)$$

12. $ratio\_label\_neighbor^T$

    C'est la proportion des voisins sociaux de i qui ont parlé de Sosh :

$$ratio\_label\_neighbor^T(i) = \frac{\sum_{j \in \mathcal{N}_s^T(i)} label^T(j)}{|\mathcal{N}_s^T(i)|}$$

13. $weighted\_ratio\_label\_neighbor^T$

    C'est la proportion des voisins sociaux de i qui ont parlé de Sosh, les termes sont pondérés par les poids des liens sociaux :

$$weighted\_ratio\_label\_neighbor^T(i) = \frac{\sum_{j \in \mathcal{N}_s^T(i)} \mathbf{S}_{ij}^T label^T(j)}{\sum \mathbf{S}_{ij}^T}$$

Nous avons calculé ces 13 variables pour les trois fenêtres suivantes : (1) 1 semaine (la semaine $t = 12$), (2) 4 semaines (les semaines $t = 9, 10, 11, 12$, ci-après "4weeks") et (3) l'ensemble de la période d'observation (depuis $t = 0$ jusqu'au $t = 12$, ci-après « all »). Nous avons donc au total 39 variables explicatives.

L'étape suivante est d'examiner la corrélation entre notre dimension latente et les variables explicatives que nous avons construites. Pour ce faire, nous avons utilisé une régression pour prédire le facteur latent à partir de ces variables explicatives. Nous avons utilisé

la régression avec *Khiops* [1] [HB07]. Khiops construit un modèle de régression avec un prétraitement (discrétisation optimale pour les variables continues et groupement optimal pour les variables catégorielles) et la sélection de variables (basé sur *Selective Naive Bayes*). Dans l'étape de sélection de variable, Khiops sélectionne un sous-ensemble de variables explicatives qui sont les plus informatives pour la variable cible.

La population à $t = 12$ est divisée selon le rapport 70%-30% pour apprentissage-test. Dans l'apprentissage, Khiops a évalué les 39 variables et trouvé que le meilleur modèle de régression contient les variables suivantes sur chaque individu :

- soc_degree^{4weeks} : le degré social de l'individu (la somme des poids des liens sociaux), calculé dans la fenêtre de 4 semaines.
- $num\_neighbor^{all}$: le nombre des voisins sociaux dans le graphe social, calculé dans toute la période d'observation (jusqu'à $t = 12$).
- $attri\_degree^{4week}$: le nombre de liens d'attribut dans le graphe d'attribut (c'est le nombre de mots uniques dans les *tweets* de l'individu), calculé dans la fenêtre de 4 semaines.
- $mean\_attri^{all}$: le poids moyen des liens d'attribut (c'est la moyenne des occurrences des mots dans les *tweets*), calculé dans toute la période d'observation (jusqu'à $t = 12$).
- $count\_label^{4weeks}$: le nombre de fois que l'individu a parlé de Sosh dans la fenêtre de 4 semaines.
- $count\_label^{all}$: le nombre de fois que l'individu a parlé de Sosh dans toute la période de l'observation (jusqu'à $t = 12$).

Pour évaluer la qualité de la régression, nous utilisons la courbe de REC (*Regression Error Characteristic*) [2] [BB03] (sur l'ensemble de test). La figure 4.6 présente la courbe de REC du meilleur modèle de régression (avec toutes les 6 variables listées au-dessus) mais aussi celles des modèles uni-variés dans lesquels une seule variable est utilisée pour prédire le facteur latent. Nous pouvons voir que le modèle de régression de toutes les variables est nettement meilleur que chacun des modèles uni-variés. C'est-à-dire notre dimension latente ne correspond pas à une seule variable mais elle est une combinaison de plusieurs variables que nous avons construites.

Prédiction avec les variables construites à la main Nous avons essayé d'utiliser l'ensemble des 6 variables explicatives listées au-dessus pour le problème de prédiction et comparer la performance avec celle du modèle à facteur latent. Nous prenons encore une fois le pas du temps $t = 12$. Plus concrètement, nous avons formé des classifieurs (naïf bayé-

1. Khiops est un logiciel de fouille développé à Orange Labs. Les détails sont disponibles à http ://www.khiops.com/.

2. Pour un modèle donné, la courbe REC trace l'estimation de la distribution cumulative de l'erreur de prédiction. L'axe horizontal de la courbe de REC représente l'erreur de prédiction et l'axe horizontal représente le pourcentage de la population. Le plus la courbe se rapproche du point (0,100%), le mieux c'est. La courbe de REC est souvent utilisée pour évaluer et comparer des modèles de régression de manière visuelle.

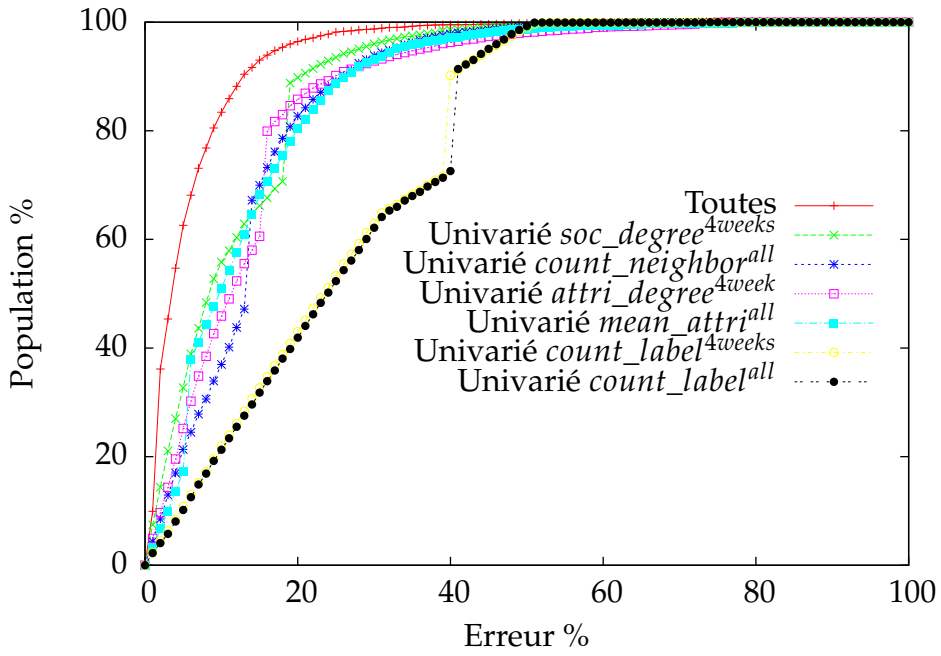


FIGURE 4.6 – Régression sur la dimension latente avec les variables explicatives construites

sien avec *Khiops Classification*) à $t = 12$ et nous les avons déployés à $t = 13$. Les classifieurs construits sont : les classifieurs uni-variés avec chacune des 6 variables explicatives listées ci-dessus et le classifieur contenant toutes ces variables. La table 4.1 présente les performances de ces modèles prédictifs, en comparaison avec le modèle contenant le facteur latent. Nous observons qu'aucune variable explicative seule ne peut prédire aussi bien que notre variable latente. Le meilleur modèle uni-varié, qui utilise la variable soc_degree^{4weeks}, donne un AUC de 0.747. Le modèle qui utilise toutes les variables explicatives donnent une performance similaire à notre modèle avec le facteur latent unique ; l'AUC obtenue par ces modèles est de l'ordre de 0.9, beaucoup plus élevé que celui des modèles uni-variés. À partir des analyses ci-dessus, nous concluons que notre facteur n'est pas réduit des variables explicatives ; il s'agit d'une combinaison de plusieurs variables. En d'autres termes, bien que notre espace latent n'ait qu'une dimension, cette dimension n'est pas quelque chose de trivial qui aurait pu être déduit directement ; elle est en fait une combinaison de plusieurs variables calculées automatiquement à partir des données (liens sociaux, attributs) dans les fenêtres temporelles de tailles différentes.

Nous avons construit les variables listées ci-dessus dans toutes les semaines et appliqué le classifieur avec ces variables pour le problème de prédiction. Cette méthode est considérée comme une méthode de référence basée sur la construction des variables explicatives à chaque pas de temps. Dans la figure 4.7, nous traçons la performance de cette méthode (*khiops avec les variables construites*), celle de notre méthode (AIMFL) et la meilleure méthode de référence basée sur les dimensions sociales "Dimensions sociales + attributs". Nous

Méthode de prédiction	AUC
Toutes les variables explicatives	**0.904**
Facteur latent	**0.901**
Univarié $attri\_degree^{4week}$	0.755
Univarié soc_degree^{4weeks}	0.747
Univarié $count\_neighbor^{all}$	0.740
Univarié $count\_label^{all}$	0.734
Univarié $count\_label^{4weeks}$	0.700
Univarié $mean\_attri^{all}$	0.683

TABLE 4.1 – Performances de prédiction (AUC) des différents modèles prédictifs en utilisant des variables explicatives (en comparaison avec le modèle contenant le facteur latent) -Modélisation à $t = 12$ et déploiement à $t = 13$

voyons que notre méthode et la méthode avec les variables construites à la main donnent des performances similaires, sauf dans quelques premières semaines. Nous remarquons ici que, dans les premières semaines, comme nous n'avons pas suffisamment de données, nous ne pouvons pas construire toutes les variables souhaitées (par exemple, dans les trois premières semaines, nous n'avons pas de variables basées sur un cumul de 4 semaines). Pour cette raison, la performance de la méthode *khiops avec les variables construites* est moins bonne dans les premières semaines.

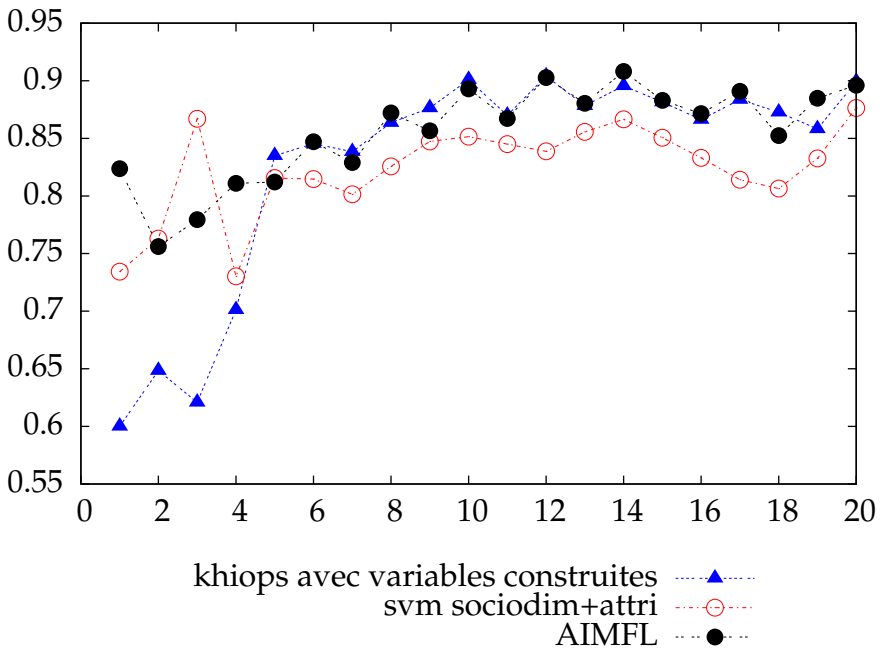


FIGURE 4.7 – Performances de prédiction (AUC) des différents modèles prédictifs

4.1.9 Discussion

Dans cette section nous avons traité un problème de prédiction réel : prédire qui parlera de la marque sur Twitter. Nous avons transformé les données collectées à partir de Twitter en forme des RSAs : les nœuds sociaux représentent les internautes sur Twitter, les liens sociaux représentent les relations de *follower* et de *co-retweet*, les attributs correspondent aux mots dans les *tweets* des internautes. Nous avons appliqué notre méthode (AIMFL) pour ce problème de prédiction. En comparaison avec les autres méthodes dont l'apprentissage est basé sur des données agrégées utilisant tout l'historique disponible, notre méthode incrémentale donne de bons résultats en termes de performance de prédiction et en termes de coût de calcul.

En analysant les dimensions latentes avec $d > 1$, nous avons trouvé que celles-ci ne sont pas indépendantes. En effet, nous avons vu qu'une seule dimension latente suffit. C'est un phénomène explicable : dans notre méthode, il n'y a aucune contrainte d'orthogonalité pour assurer que les dimensions latentes soient indépendantes. Ces dimensions latentes peuvent être corrélées entre elles et c'est le cas ici si l'on choisit a priori un nombre supérieur au nombre suffisant. Ainsi on ne peut pas améliorer la performance en augmentant systématiquement le nombre de dimensions latentes : choisir a priori d très grand n'est pas la meilleure stratégie, d'autant plus que la complexité est en $O(d^3)$ et nuit aux temps de calcul. Il y a ainsi un nombre au-delà duquel on introduit de la redondance. D'après nos expériences non décrites ici, ceci n'est pas un problème – si l'on s'affranchit du temps de calcul – car les variations de d n'engendrent pas de grandes variations de performance.

Notons que même avec une seule dimension latente, notre méthode de réduction de dimension donne une très bonne performance de prédiction associée à SVM sur les données de Twitter. Comme nous l'avons montré expérimentalement, la dimension latente détectée est une combinaison non triviale de plusieurs variables, et chose intéressante, ces variables ont été calculées sur des fenêtres temporelles différentes.

Nous avons aussi comparé notre méthode avec l'apprentissage supervisé avec les variables explicatives construites à la main. Notre méthode donne des performances équivalentes avec l'apprentissage supervisé avec les variables construites à la main (voir figure 4.7). L'avantage de notre méthode, par rapport aux méthodes basées sur la construction des variables à la main, est qu'elle peut trouver les variables latentes informatives et les mettre à jour de manière incrémentale.

4.2 Prédiction d'actes commerciaux des clients

Pour des raisons de confidentialité, ces travaux ont été présentés au jury de thèse via une annexe confidentielle et ne peuvent pas être rendus publics.

CHAPITRE 5

CONCLUSION

Sommaire

5.1 Bilan

Dans cette thèse, nous nous sommes intéressés aux nouvelles techniques de fouilles de données dans le contexte de la gestion de la relation client intercanale. L'objectif principal de la thèse est de prédire les comportements des clients dans le cadre d'une stratégie de relation client intercanale.

Nous avons d'abord effectué une analyse des besoins (en termes de fouille de données) pour la stratégie de relation client intercanale. Cette analyse nous a permis de dégager les problématiques principales de la thèse : l'exploitation simultanée des données attribut-valeur et les données de type relation (i.e les données relationnelles), l'intégration des contenus générés sur les média sociaux par les utilisateurs (données creuses à grande dimension), la prise en compte de l'aspect dynamique des données et la capacité de passage à grande échelle.

L'état de l'art (le Chapitre 2) a présenté différentes techniques de fouille de données pour le CRM et le Social CRM. Les techniques de la fouille de données classiques comme la classification, la régression et le clustering se trouvent souvent dans les systèmes de CRM classiques, et traitent des données tabulaires. Les techniques de fouille de média sociaux sont aussi souvent utilisées dans le Social CRM, dans les applications connues comme les mesures de l'influence, le marketing viral et le monitoring des média sociaux. Nous avons aussi examiné les techniques d'apprentissage statistique relationnel, une famille de techniques d'apprentissage qui permettent de combiner les données tabulaires et les relations sociales. Après cette étude bibliographique, nous avons identifié certaines approches applicables pour l'objectif de la thèse : la classification supervisée (avec les variables explicatives à construire à partir des données intercanales), le modèle de l'influence (dont la méthode de voisinage est une version simple) et les méthodes basées sur les dimensions latentes [TL11]. Ces méthodes ont été utilisées comme les méthodes de référence dans nos expérimentations.

Pour attaquer les problématiques posées dans la thèse, nous avons proposé une nouvelle approche basée sur les modèles à facteurs latents (Chapitre 3). D'abord, nous représentons les données provenant de plusieurs canaux sous forme de *réseau social attribué* (RSA). Un réseau social attribué est un réseau social avec des nœuds représentant des individus et liens entre ces nœuds ; ce réseau a la particularité d'être enrichi par des nœuds d'attribut, un deuxième type de nœuds qui caractérisent les individus ; des liens entre les différents types de nœuds matérialisent cette caractérisation. Cette représentation a pour but de représenter les données de CRM intercanales dans lesquelles nous trouvons nos clients caractérisés par des profils client (données tabulaires issues du SI encodées dans des attributs SI), les relations sociales entre eux, les contenus qu'ils ont générés (attributs sociaux).

Pour prendre en compte la dynamique de données, nous utilisons une séquence de RSAs, chacun représente les données récente à un pas de temps. Nous introduisons ensuite

une nouvelle méthode d'apprentissage incrémental basée sur les modèles à facteurs latents. L'idée est de projeter les nœuds du graphe attribué dans un espace latent de faible dimension. L'apprentissage des facteurs latents est effectué via la *factorisation de matrice*. Nous utilisons une adaptation de l'algorithme des *moindres carrées en alternance* pour la factorisation de matrice. Notre méthode est capable d'effectuer l'apprentissage incrémental - à chaque pas de temps il prend la partie de données courantes pour mettre à jour les facteurs latents des nœuds. Les facteurs latents appris sont ensuite utilisés pour l'apprentissage d'un classifieur (SVM) pour prédire les comportements des clients.

Nous avons testé la méthode d'abord sur les jeux de données intercanales synthétiques. Nous avons développé pour cela un générateur de données synthétiques pour simuler les données intercanales issues de deux canaux : le SI client (les données clientèle) et les média sociaux. Dans le cas général où toutes les parties des données (le graphe social, les attributs sociaux, les variables du SI) sont informatives, la méthode proposée donne une performance comparable (en termes d'AUC) avec les méthodes de référence (notamment la méthode utilisant l'apprentissage supervisé et les dimensions latentes [TL11]). Nous avons aussi examiné plusieurs scénarios, dans lesquels nous réglons les dépendances statistiques entre la variable cible (qui indique les comportements des clients à prédire) et différentes parties de données. Nous concluons que notre méthode est comparable, voir meilleure que les méthodes de référence (apprentissage supervisé en utilisant conjointement graphe social, les attributs et les variables du SI) dans le cas où le graphe social est suffisamment informatif et possède une caractéristique d'homophilie.

Nous avons ensuite appliqué notre méthode sur les données réelles (Chapitre 4). Dans la première application, nous essayons de prédire qui parlera de la marque (Sosh) sur Twitter. Le jeu de données a été collecté via Twitter API, il concerne les *followers* de Sosh sur Twitter. Nous avons montré que notre méthode peut donner des performances de prédiction comparables avec les méthodes de référence, y compris la classification supervisée avec les variables explicatives construites à la main. Dans la deuxième application, nous essayons de prédire un type d'acte commercial des clients à partir des données intercanales réelles. Nous avons comparé notre méthode avec les méthodes de référence, notamment les méthodes utilisant Khiops avec les variables explicatives construites à la main.

5.2 Apports applicatifs de la thèse

En termes applicatifs, la thèse a des contributions dans le cadre d'une stratégie de relation client intercanale et pour la caractérisation de l'engagement des clients avec l'entreprise.

– Dans le cadre d'une stratégie de relation client intercanale, les apports de la thèse sont les études des techniques permettant de combiner les données issues de multiples ca-

naux pour l'apprentissage. La première contribution de la thèse est l'analyse des besoins dans laquelle nous avons identifié les grands défis de la fouille de données dans une stratégie de relation client intercanale. À partir de cette analyse, nous avons effectué un état de l'art sur les travaux académiques connexes. Ensuite, nous nous concentrons sur l'exploitation de données relationnelles et les données à grande dimension (dans les média sociaux), conjointement avec les données tabulaires (le SI client). Pour ce faire, nous proposons d'utiliser un graphe social attribué pour représenter les données et de développer un algorithme d'apprentissage basé sur les facteurs latents.

– Concernant l'engagement, sa définition et son exploitation figurent dans l'objectif principal et nous ont servi de fil conducteur : étudier les techniques permettant de caractériser et prédire les comportements des clients - les comportements qui reflètent l'engagement du client dans l'entreprise. Dans les travaux expérimentaux, nous avons traité les problèmes de prédiction intéressants pour une entreprise : prédire qui parlera de Sosh sur Twitter et un type d'acte commercial avec les données intercanales fournies par Orange. Ce ne sont que des exemples de comportements clients, un type d'activité sur les media sociaux et un type d'acte commercial. Nous proposons dans notre travail d'utiliser les facteurs latents pour prédire ce type de comportements. Ces comportements prédits sous forme de variables cibles, pourront alors participer au calcul d'un score d'engagement par le service de gestion de la relation client.

Cependant, les travaux expérimentaux avec les données réelles de l'entreprise ont montré que la méthode de prédiction proposée n'apporte pas de gain en termes de performance (AUC) par rapport aux méthodes classiques basées sur la construction des variables explicatives caractérisant explicitement les interactions sociales. En l'état actuel, la méthode proposée n'est pas pertinente dans le contexte applicatif de la thèse (au moins avec les données intercanales que nous avons testées). Nous avons réfléchi aux différentes pistes d'amélioration de la méthode, notamment d'autres modèles à facteurs latents permettant d'exploiter différents types de corrélations entre les individus dans le graphe social (cf. section 5.5). Nous avons identifié également d'autres champs applicatifs plus généraux qui pourraient être intéressants pour l'entreprise, comme par exemple dans un futur proche, les données volumineuses et hautement variables de l'Internet des Objets (cf. section suivante).

5.3 Apports académiques de la thèse

Plus largement, et hors le contexte spécifique applicatif du passage au CRM intercanal d'une grande entreprise, la thèse présente des contributions.

Bien que la méthode proposée ne puisse pas garantir dans tous les cas une meilleure performance que les méthodes classiques (celles utilisées actuellement et basées sur la construc-

tion des variables à la main), son avantage est la capacité de trouver de manière automatique les facteurs latents informatifs à la prédiction. Ces variables latentes constituent une représentation de dimension réduite des données complexes contenant à la fois les interactions sociales, les attributs et les contenus à grande dimension (les textes).

De part nos expériences sur nos données synthétiques, nous avons montré que notre méthode assure de bonnes performances dans le cas où le graphe social est suffisamment informatif et que le graphe social possède une caractéristique d'homophilie. Dans ce contexte, nous pensons que notre méthode est particulièrement utile dans le cas où il est difficile de chercher des variables explicatives informatives. Dans les médias sociaux les données sont riches, la tendance Big Data est de tout collecter de manière non structurée. Notre méthode qui apprend automatiquement la représentation (de faible dimension) à partir de données brutes et exploite cette représentation pour la prédiction peut prendre ici tout son intérêt. Ceci sera d'autant plus problématique avec l'internet des objets, où les médias sociaux seront interconnectés dans un futur proche avec pléthore d'objets non connus à l'avance, introduisant une grande variabilité des variables explicatives dans les méthodes de prédiction, ce qui ouvre d'autres champs applicatifs pertinents, notamment pour une entreprise comme Orange .

En termes de coût de calcul, nous avons montré que la complexité de notre algorithme d'apprentissage est *linéaire* en fonction de la taille de données en entrée. Notre méthode est donc capable de passer à grande échelle. À noter que la complexité est d'ordre d^3 (d est le nombre de dimensions latentes), et que selon les jeux de données, d peut prendre des valeurs relativement grandes (environ 50 sur nos expérimentations sur les données intercanales fournies par Orange) comme très faibles (1 pour nos expérimentations sur Twitter). Cependant, l'algorithme d'apprentissage des facteurs latents est parallélisable. Dans ce travail, nous avons en effet implémenté une version parallélisée de l'algorithme.

Dans cette thèse, nous avons aussi développé un générateur de données synthétiques pour générer les données issues de deux canaux : les média sociaux et le SI client. Bien que ce générateur de données s'appuie sur un modèle très simplifié des données issues des media sociaux et du SI client, nous avons réussi à l'utiliser pour générer les données dans certains scénarios pour inspecter les comportements de notre méthode. Le générateur peut être étendu pour tester autres scénarios dans la fouille de données intercanales, ou même au de-là du contexte de la thèse (par exemple, pour des études des graphes sociaux attribués).

5.4 Limitations

En pratique, notre méthode d'apprentissage a un inconvénient concernant le réglage des paramètres. Nous devons effectuer une procédure de réglage des paramètres sur les don-

nées de validation pour trouver les valeurs optimales des paramètres. Cette procédure qui consiste à chercher les valeurs optimales des paramètres sur une grille des valeurs prend beaucoup de temps.

Une autre limitation de notre approche est qu'il n'y a pas une interprétation intuitive des facteurs latents. Contrairement aux variables explicatives construites à la main, un facteur latent ne représente pas des caractéristiques explicites sur les individus. Un facteur latent correspond à une caractéristique latente (non-observée) des individus, qui de plus peut être une combinaison non-intuitive des plusieurs variables provenant de différentes parties des données (se référer à l'analyse des facteurs latents des expérimentations sur les données Twitter dans la section 4.1.8). Dans les cas où on veut une interprétation du modèle prédictif (e.g pour identifier quelles parties de données sont informatives, quelles parties correspondent au bruit), notre méthode à base de facteurs latents n'est pas appropriée.

Dans la thèse, nous n'avons pas répondu à toutes les problématiques identifiées au début de la thèse. Nous aurions souhaité étudier la dynamique des données, en particulier les données issues des média sociaux. La méthode d'apprentissage incrémental des facteurs latents est adaptée à la dynamique de données, dans le sens où seules les données courantes sont utilisées à chaque pas de temps pour mettre à jour les facteurs latents. Les nouveaux contenus sont pris en compte (en créant les nouveaux nœuds d'attribut à chaque pas de temps). Pourtant, la réponse de notre méthode aux dynamiques du modèle, par exemple l'impact de la taille d'incrément (par exemple le nombre de nouveaux nœuds sociaux, de nouveaux nœuds d'attribut, etc.) n'a pas encore été étudiée. De plus, les données issues des média sociaux reflètent souvent des événements éphémères, l'évolution des données de média sociaux n'est pas forcément dans la même temporalité que celles des autres canaux. Cette caractéristique dynamique est intéressante et nous ne l'avons pas encore considérée.

5.5 Perspectives

Dans les perspectives, nous souhaiterions aussi tester d'autres approches d'apprentissage basées sur les facteurs latents. Dans cette thèse, nous nous sommes basés principalement sur l'hypothèse d'*homophilie* du graphe social (hypothèse que les individus connectés partagent des comportements communs). Pour exploiter l'homophilie, nous utilisons une approche de la factorisation de matrice (*factorisation régularisée relationnelle de matrice* - FRRM). En réalité, les données sociales, en particulier, les forums d'entraide, peuvent posséder d'autres types de corrélation entre les individus dans les réseaux sociaux, par exemple l'*équivalence stochastique* (cf. la section 2.3.1). Dans ces cas, d'autres types de factorisation de matrice seraient probablement plus adaptés, par exemple la *factorisation collective de matrices* [SG08]. Nous avons présenté cette approche (Chapitre 3) mais elle n'a pas été testée dans les

expérimentations menées dans cette thèse.

Nous pensons aussi à améliorer la procédure de validation pour identifier les paramètres optimaux pour notre modèle. La procédure de validation que nous avons utilisée est une recherche exhaustive sur une grille des valeurs des paramètres. Cette procédure est coûteuse en temps de calcul. Par une étude théorique plus approfondie sur les impacts des paramètres, nous pouvons probablement réduire l'espace de recherche des valeurs optimales des paramètres.Une autre approche pour faciliter le réglage des paramètres est d'utiliser un jeu de données de validation de petite taille. Le défi ici est de constituer un jeu de données de petite taille mais toujours représentatif des données originales.

Nous souhaiterons également approfondir la prise en compte de la dynamique dans notre méthode. Nous envisageons d'abord de tester la méthode sur différents jeux de données avec différents caractères dynamiques. Par exemple, nous pouvons rajouter un caractère dynamique à la variable cible (les comportements des clients) comme suit : par exemple, considérer le cas où seules les données les plus récentes sur les média sociaux sont informatives pour prédire la variable cible. Le taux d'apparition de nouveaux nœuds sociaux et de nouveaux attributs sont les types de dynamique à considérer. Ces expérimentations peuvent être menées sur les jeux de données synthétiques générées (avec le générateur que nous avons développé) et éventuellement les jeux de données réelles si disponibles. Nous pouvons aussi faire varier la taille d'incrément (le nombre de nouveaux noeuds, de nouveaux attributs à chaque pas de temps) et voir l'impact sur la performance de prédiction. Les pistes d'amélioration de la prise en compte de la dynamique sont aussi envisagées. Par exemple, basée sur l'hypothèse « les données récentes sont plus pertinentes », nous pouvons améliorer la représentation du réseau social attribué en ajoutant une pondération temporelle sur les liens.

Nous considérons aussi les techniques de la fouille d'opinions dans les travaux futurs. Les opinions positives (ou négatives) des clients sur les média sociaux vis-à-vis de la marque sont des indicateurs intéressants de l'engagement. La prédiction des opinions positives ou négatives des clients sur les média sociaux (à l'échelle individuelle) est donc intéressante pour le Social CRM. De plus, la polarité de l'opinion d'un client peut être utilisée comme une variable explicative pour prédire son comportement.

Dans cette thèse, nous nous intéressons à prédire des comportements des clients. Pour ce faire, les facteurs latents appris par notre algorithme sont utilisés pour la classification supervisée. Dans une autre perspective, les facteurs latents appris sur un graphe attribué peuvent être utilisés d'une autre manière, pour d'autres problèmes. Dans notre travail publié à EGC en 2014 [LTBC14], nous utilisons les modèles à facteurs latents pour prédire des attributs sur les nœuds dans un réseau social attribué. Ici, le problème de prédiction d'attributs est formulé comme la prédiction des liens d'attribut. La méthode a été appliquée

sur les données de BlogCatalog pour prédire les groupes d'intérêts des blogueurs (voir Annexe C pour les détails). La même idée peut être appliquée dans le cadre d'une stratégie de relation client intercanale, avec objectif par exemple de prédire des produits ou services qui pourraient être intéressants pour un client. Bien que ce problème a été considéré depuis longtemps (la recommandation personnalisée), le point innovant ici est l'utilisation les données intercananales (combinaisons de données sociales et données tabulaires).

ANNEXE A

LE JEU DE DONNÉES SYNTHÉTIQUE

t	Nombre de nœuds sociaux	Nombre de nœuds d'attribut	Nombre de liens sociaux	Nombre de liens d'attributs
0	100 000	27218	388046	100940
1	110 000	29545	183354	171807
2	120 000	32207	202158	266185
3	130 000	35432	224134	329437
4	140 000	38315	244324	368647
5	150 000	40383	265438	399079
6	160 000	43512	286972	426657
7	170 000	46195	307318	454360
8	180 000	48991	327224	481823
9	190 000	51949	348216	508674

TABLE A.1 – Le jeu de données synthétique : les RSAs générés dans chaque période

ANNEXE B

LE JEU DE DONNÉES TWITTER

B.1 Découpage de données en semaines

Semaine	Nombre d'individus	Nombre de tweets	Nombre de retweets	Nombre de liens follower
0	21620	124980	93377	299648
1	21960	133222	95724	385492
2	22316	130032	96812	397244
3	22679	129619	99991	408672
4	23003	124978	91406	422408
5	23035	136108	102541	428018
6	23206	169784	127012	428018
7	24731	160233	123035	483364
8	25063	157396	115743	501102
9	26234	165352	117619	535472
10	27030	170344	127353	562280
11	27394	170217	126643	583716
12	27766	167782	134655	601524
13	28276	173811	132006	623176
14	28628	179314	130825	641014
15	28806	211943	169547	651864
16	29182	232334	181633	675292
17	29501	203730	143839	690760
18	29718	208446	154131	703122
19	30142	213838	148213	724170
20	30400	221001	153501	735734

TABLE B.1 – Les 21 sous-jeux de données

Semaine	Nombre de nœuds sociaux	Nombre de nœuds d'attribut	Nombre de liens sociaux	Nombre de liens d'attributs
0	21620	56124	421356	300189
1	21960	87372	621584	302468
2	22316	112999	640302	298862
3	22679	135750	674008	293524
4	23003	155986	686924	285729
5	23035	175940	702692	305291
6	23206	196995	773920	340660
7	24731	218027	866154	365339
8	25063	238550	885250	372256
9	26234	259208	1014520	391815
10	27030	279731	1085600	405430
11	27394	299680	1020314	403164
12	27766	319398	991598	406817
13	28276	338781	1151486	417986
14	28628	358090	1230586	430333
15	28806	376742	1110240	437719
16	29182	396247	1241674	456614
17	29501	411789	1233702	365312
18	29718	426891	1045148	369675
19	30142	431535	1062810	419241
20	30400	446658	1052272	377367

TABLE B.2 – Les RSAs construits à chaque semaine

ANNEXE C

APPRENTISSAGE INCRÉMENTAL DES FACTEURS LATENTS POUR LA PRÉDICTION DES ATTRIBUTS DANS UN RÉSAU SOCIAL ATTRIBUÉ

Incremental learning with latent factor models for attribute prediction in social-attribute networks

Duc Kinh Le Tran*,** Cécile Bothorel*
Pascal Cheung Mon Chan**

*UMR CNRS 3192 Lab-STICC
Département LUSSI – Télécom Bretagne
{duc.letran, cecile.bothorel}@telecom-bretagne.eu
**Orange Labs
{duckinh.letran, pascal.cheungmonchan}@orange.com

Abstract. In this paper, we are interested in the problem of predicting attributes on the nodes in a social network. Most of the existing techniques addressing this problem are offline learning techniques and are not suitable in situations where massive data come in stream like social media. In this work, we use *latent factor models* to predict unknown attributes of the nodes in a social network and propose a method to incrementally update the prediction model on the arrivals of new data. Experiments on a real social media dataset show that our method is more rapid and can guarantee acceptable performances in comparison with state-of-the-art non-incremental techniques.

1 Introduction and problem statement

With the explosion of social media on the Internet in recent years, mining social media content has become more and more critical for many domains. One of the challenges of mining social media is how to leverage *relational information* (e.g friendships, interactions between social media users) and simultaneously *attributes* (e.g. users' interests, textual or any other additional information). Another challenge lies in the fact that these media provide vast and continuous streams of data. Using offline learning techniques, we have to aggregate all the data available from the past until the present. This approach is not suitable in this situation because (1) as new data come, the size of the dataset grows, it get more and more expensive to learn and to apply the model (2) this approach cannot capture the dynamic of the data stream: old data and recent data are treated uniformly.

In this paper, we address both challenges by introducing an incremental learning method for the task of predicting attributes of social actors in a social network. This problem has many real world applications, for example to predict users' interests or hobbies using social media. We build a graph of interactions among the social media users and enrich the graph with a set attributes on nodes. As the data (nodes, links) arrive as a permanent stream, we want to build models to periodically predict unknown attributes on the nodes.

To formulate our problem, we adopt the *social-attribute network* (Yin et al. (2010)). A *social-attribute network (SAN)* contains a social network G_s=(V_s, E_s) where V_s is the set of

Incremental learning with LFM in social-attribute networks

nodes and E_s is the set of edges. The social graph is augmented with a bipartite graph $G_a = (V_s \cup V_a,\, E_a)$, called the attribute graph, connecting the *social nodes* in V_s with *attribute nodes* in V_a. The edges in E_s are *social links* and the edges in E_a (connecting social nodes and attribute nodes) are *attribute links.* There are 2 types of attribute link between a social node i and an attribute node k: a *positive link* if i has the attribute k; a *negative link* if i doesn't have k and in case we don't know whether i has k or not, there is no link between i and k (missing link). We are interested in the problem of predicting attributes of nodes (i.e the nature - positive or negative - of missing links in the attribute graph G_a) in the context of incremental learning. In this context, at each time step t, we have a snapshot $\mathcal{G}(t)$ of the SAN which represents all data (nodes and links) available from the past until t. In comparison with the previous snapshot $\mathcal{G}(t-1)$, $\mathcal{G}(t)$ has new nodes and new links. The new nodes can be social nodes or attribute nodes (in the experiments we only consider new social nodes due to limitation of the data set). We denote by $\Delta\mathcal{G}(t)$ the SAN constituted of the new links which have just been added at the time step t. The SAN $\mathcal{G}(t)$ contains two components (sub-networks), the first component is the snapshot $\mathcal{G}(t-1)$, and the second component is $\Delta\mathcal{G}(t)$. We formulate our incremental learning problem as follows: assume that we have built a model M_{t-1} to predict attributes of nodes at the time step $t-1$, *our problem is to update the model M_{t-1} with new data (nodes and links in $\Delta\mathcal{G}(t)$) to predict unknown attributes of nodes at the time step t.*

In the followings, we review *latent factor models* and *matrix factorization* in batch learning (Section 2) and then propose an approach for incremental learning based on these techniques (Section 3). We present some encouraging experimental results in Section 4. Finally in Section 5 we conclude and point out some promising directions in future work.

2 Latent factor model and matrix factorization

As stated earlier, the learning approach proposed in this paper is inspired from *latent factor models (LFM)* (Bartholomew et al. (2011)), which have long been used in statistics and machine learning. A LFM is a statistical model that represents each data instance by a set of latent variables. *Matrix factorization* (MF) can be considered as a method of latent factor modeling in which latent variables are continuous. The basic idea of MF is to decompose a high dimensional data matrix into lower dimensional matrices.

Techniques of MF have also been extended to handle multiple matrices at a time. Singh and Gordon (2008) introduced *collective matrix factorization (CMF)*. CMF can deal with relational data in which there are many types of entity and many types of relation between entities, each type of relation is represented by a relational matrix. CMF tries to map entities into a common latent space by factorizing simultaneously multiple relational matrices. In our problem setting, we have two matrices : the adjacent matrix of the social network (denoted by S) and the attribute matrix (denoted by A where A_{ik} is a binary value indicating whether the attribute link $(i,\, k)$ is positive or negative). Using CMF, we minimize:

$$\begin{aligned} Q_{CMF}(U,\, P,\, \mathcal{G}) =& \alpha \sum_{(i,\, j) \in E_s} \left(S_{ij} - u_i u_j^T\right)^2 + \sum_{(i,\, k) \in E_a} \left(A_{ik} - u_i p_k^T\right)^2 \\ &+ \lambda \left(\sum_{i=1}^{n_s} \|u_i\|^2 + \sum_{k=1}^{n_a} \|p_k\|^2 \right) \end{aligned} \tag{1}$$

where E_s is the set of social links, E_a is the set of attribute links; U is the matrix constituted of the latent vectors of all the social nodes and similarly, P is the matrix constituted of the latent vectors of all the attribute nodes of $\mathcal{G}$. The parameter α allows to adjust the relative importance of the social network in the model. The third term is a regularization term to penalize complex models with large magnitudes of latent vectors. λ is a regularization parameter.

Li and Yeung (2009) proposed another extension of MF, called *relation regularized matrix factorization* (*RRMF*). RRMF simultaneously exploits the social graph and the attribute graph by minimizing (with the same notations as in Equation 1):

$$\begin{aligned} Q_{RRMF}\left(U,\, P,\, \mathcal{G}\right) =& \alpha \sum_{(i,\, j)\in E_s} S_{ij} \left\| u_i - u_j \right\|^2 + \sum_{(i,\, k)\in E_a} \left(A_{ik} - u_i p_k^T\right)^2 \\ &+ \lambda \left(\sum_{i=1}^{n_s} \left\| u_i \right\|^2 + \sum_{k=1}^{n_a} \left\| p_k \right\|^2 \right) \end{aligned} \tag{2}$$

We can see that this is in fact the factorization of the attribute matrix A when adding regularization term $\alpha \sum_{(i,\, j)\in E_s} S_{ij} \left\| u_i - u_j \right\|^2$. This term is called the *relational regularization term* which allows to minimize the distances between connected social nodes in the latent space. The RRMF approach assumes that connected social actors tend to have similar profiles.

3 Incremental learning with latent factor models

In the incremental learning context defined in Section 1, we need to learn a prediction model (i.e the latent features of nodes) at each time step. The *batch learning* approach suggests that we learn the latent features at each time step using the whole snapshot $\mathcal{G}\left(t\right)$

$$U^\star\left(t\right),\, P^\star\left(t\right) = \underset{U,\, P}{\arg\min}\, Q\left(U,\, P,\, \mathcal{G}\left(t\right)\right) \tag{3}$$

where Q is one of the two objective functions defined above (Equation 1 and Equation 2). Different from the batch learning method, the incremental method learns a model only from new data (i.e SAN $\Delta\mathcal{G}\left(t\right)$) when reusing the old model, i.e latent features of nodes calculated in the previous time step. To do this, we minimize the following objective function:

$$\begin{aligned} Q_{inc}\left(U,\, P,\, t\right) =& Q\left(U,\, P,\, \Delta\mathcal{G}\left(t\right)\right) \\ &+ \mu \left(\sum_{i\in V_s(t-1)} \left\| u_i - u_i^\star\left(t-1\right) \right\|^2 + \sum_{k\in V_a(t-1)} \left\| p_k - p_k^\star\left(t-1\right) \right\|^2 \right) \end{aligned} \tag{4}$$

where $V_s\left(t-1\right)$ and $V_a\left(t-1\right)$ are respectively the set of social nodes and the set of attribute nodes in the previous time step; $u_i^\star\left(t-1\right)$ and $p_k^\star\left(t-1\right)$ are respectively the latent vectors of the social node i and the attribute node k learned in the previous time step and μ is a parameter of the model. This objective function consists of two terms. The first term is the objective function of MF on the incremental graph $\Delta\mathcal{G}\left(t\right)$. The second term is a regularization term for minimizing the shifts of latent features of the same nodes between time steps. By minimizing the two terms simultaneously, we learn latent features of nodes both from the new data and

from the latent features of existing nodes of the previous time step. We can easily see that the latent features of an existing node are updated if and only if there are new links connecting to it. The parameter μ allows to tune the contribution of the previous model to the current model.

In terms of optimization, we adapt the *Alternating Least Squared* (Zhou et al. (2008)) algorithm to minimize Q in Equation 3 for batch learning or Q_{inc} in Equation 4 for incremental learning. The basic idea of this algorithm is to solve the least square problem with respect to the latent features of one node at a time until convergence. The complexity of the algorithm linearly depends on the number of squared terms in the objective function, which is the total number of nodes and number of links in the SAN. In other words, the learning algorithm has linear complexity with respect to the size of the data. In case of incremental learning, when optimizing only on recent data ($\Delta\mathcal{G}(t)$), we can gain a lot in terms of computational cost.

4 Experiments

4.1 Experimental setup

The dataset used in these experiments is BlogCatalog, collected and used by Tang and Liu (2011). In BlogCatalog [1], a blogger can specify his connections with other bloggers. In addition, when submitting a new blog, a blogger specifies the categories of the blog among a set of pre-defined categories. A blogger's interests can be inferred from the categories of his blogs. The dataset contains only a small portions of the whole network: 10312 bloggers, 333983 connections between bloggers, 39 categories, and each blogger has on average 1.4 categories of interest. We can build a SAN out of this data set where bloggers are social actors and categories are attributes. Since we don't have a real data stream, we construct artificial SAN snapshots from this static data set to test our incremental learning method. We build SAN snapshots at 6 time steps in our experiment. We only consider adding new social nodes at each time step (the set of 39 attribute nodes is fixed). We initially pick 50% of the total social nodes and build the SAN snapshot $\mathcal{G}(0)$ from these nodes and all links (social links and attribute links) that involve them. At each time step $t \in \{1, 2, 3, 4, 5\}$, we randomly take 10% of the total social nodes (only nodes which have not been taken yet). We add these social nodes and all their social links to build a new snapshot $\mathcal{G}(t)$. About the attribute links, we assume that the attributes of new nodes at t are unknown until the next time step $t + 1$.

Our objective is to predict unknown attributes of nodes with our incremental methods (*Incremental CMF*, *Incremental RRMF*) at each time step. We compare our incremental learning methods with the *batch learning* approach (i.e using the whole snapshot $\mathcal{G}(t)$ at each time step t). Three batch learning methods are used to compare: batch learning with CMF, RRMF and another state-of-the-art method called *Social Dimension (SocialDim)* (Tang and Liu (2011)). The basic idea of this method is to transform the social network in to features of nodes using a graph clustering algorithm (where each cluster, also called a called a *social dimension*, corresponds to a feature) and then train a discriminative classifier (Support Vector Machine (Cortes and Vapnik (1995))) using these features. It has been shown that the SocialDim outperforms other well-known methods of classification in a network.

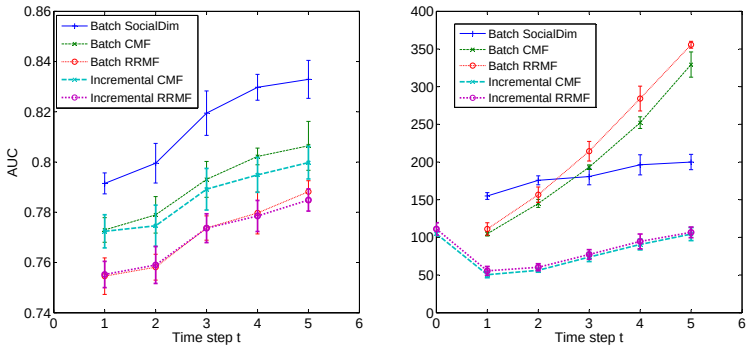
To measure the performances of the different prediction methods, we use *Area Under ROC Curve* (AUC) (Bradley (1997)). At each time step, the AUC is computed from the prediction

1. http://www.blogcatalog.com/

scores and the true labels of all missing links. We also measure the computational time of each method to show empirical gain in complexity of our incremental method.

About the choices of parameters, we have observed that the performances of LFM methods are relatively stable with changes of λ, α and μ in both batch learning and incremental learning. We have set $\lambda = 1.0, \alpha = 1.0, \mu = 100$ for CMF and $\lambda = 1.0, \alpha = 1.0, \mu = 10$ for RRMF to produce representative results for each methods in our experiment. The number of latent factors is set to 50, at which CMF and RRMF attain their maximal stable performances.

4.2 Experiment results



(a) AUC (b) Learning time

FIG. 1: Performance and learning time of incremental learning compared with batch learning

We perform 5 runs and plot the average AUC of each method in each time step in Figure 1a. We observe that the incremental learning techniques cannot give better performances than batch learning methods in all time steps. This is an expected observation: in this experimental setting, the "data stream" is not real. However, both CMF and RRMF give almost the same performance in batch learning and incremental learning (in all cases the difference is not more than 1%). In other words, with LFM, we can incrementally learn the prediction model instead of learning from scratch without any significant loss in performance. When comparing CMF and RRMF, we see clearly that CMF is better. We can also see that the performances of our incremental learning techniques are not too far from those of the reference method - SocialDim (difference of 4% in the worst cases).

Figure 1b shows the learning time (in seconds) of each tested method. To be fair, all the methods are implemented and executed in MATLAB on the same machine (CPU 2.5GHz and 4GB of RAM). The incremental learning techniques require to learn a model from the SAN $\mathcal{G}(0)$ (without prediction) at the time step 0, while the batch learning techniques don't need this step. But in the subsequent time steps (1 to 5), the incremental techniques always have much smaller learning time than that of the batch learning methods. In batch learning, the learning time of CMF and RRMF increases rapidly after each time step. Although the learning time of SocialDim increases less rapidly than that of CMF and RRMF, it is still very long compared to our incremental methods.

Incremental learning with LFM in social-attribute networks

5 Conclusion

Motivated by the challenges of social media mining, we have proposed an incremental learning method based on LFM. Two alternatives (CMF and RRMF) inspired from LFM have been tested for the problem of incremental attribute prediction in a social network. Our learning algorithm can achieve relatively good performance compared to the reference method based on Social Dimension, a non-incremental classification method. In future work, we will test our incremental approach on real data streams. We expect that our incremental learning method can capture the dynamic of data stream and give better performances than batch learning. We also consider possible extensions of our models to deal with more complex data in social media, for example to consider other types of nodes and links in the SAN, to include attributes on edges, to handle directed links, etc.

References

Bartholomew, D. J., M. Knott, and I. Moustaki (2011). *Latent Variable Models and Factor Analysis: A Unified Approach*. Wiley Series in Probability and Statistics. Wiley.

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recogn. 30*(7), 1145–1159.

Cortes, C. and V. Vapnik (1995). Support-vector networks. *Machine learning 20*(3), 273–297.

Li, W. and D. Yeung (2009). Relation regularized matrix factorization. In *IJCAI-09*, IJCAI'09, pp. 1126–1131. Morgan Kaufmann Publishers Inc.

Singh, A. and G. Gordon (2008). Relational learning via collective matrix factorization. In *Proceeding of the 14th ACM SIGKDD*, Number June, pp. 650–658. ACM.

Tang, L. and H. Liu (2011). Leveraging social media networks for classification. *Data Mining and Knowledge Discovery 23*(3), 447–478.

Yin, Z., M. Gupta, T. Weninger, and J. Han (2010). A Unified Framework for Link Recommendation Using Random Walks. ASONAM '10, pp. 152–159. IEEE Computer Society.

Zhou, Y., D. Wilkinson, R. Schreiber, and R. Pan (2008). Large-Scale Parallel Collaborative Filtering for the Netflix Prize. In *Proceedings of the 4th international conference on Algorithmic Aspects in Information and Management*, AAIM '08, pp. 337–348. Springer-Verlag.

Résumé

Dans ce travail, nous nous intéressons au problème de la prédiction d'attributs sur les nœuds dans un réseau social. La plupart des techniques sont hors ligne et ne sont pas adaptées à des situations où les données arrivent massivement en flux comme dans le cas des médias sociaux. Dans ce travail, nous utilisons les modèles de variables latentes pour prédire les attributs inconnus des nœuds dans un réseau social et proposer une méthode pour mettre à jour incrémentalement le modèle avec des nouvelles données. Des expérimentations sur un jeu de données issues des médias sociaux montrent que notre méthode est moins coûteuse en temps de calcul et peut garantir des performances acceptables en comparaison avec les techniques non-incrémentales de l'état de l'art.

BIBLIOGRAPHIE

[AMK04] C. ARCHAUX, A. MARTIN et A. KHENCHAF : An SVM based churn detector in prepaid mobile telephony. *Proceedings. 2004 International Conference on Information and Communication Technologies : From Theory to Applications, 2004.*, 2004.

[APY02] Charu C. AGGARWAL, Cecilia PROCOPIUC et Philip S. YU : Finding localized associations in market basket data. *IEEE Transactions on Knowledge and Data Engineering*, 14(1):51–62, 2002.

[AXV+11] Apoorv AGARWAL, Boyi XIE, Ilia VOVSHA, Owen RAMBOW et Rebecca PASSONNEAU : Sentiment analysis of twitter data. *In Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 30–38, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[BA99] Albert-Laszlo BARABASI et Reka ALBERT : Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[BB03] Jinbo BI et Kristin P. BENNETT : Regression error characteristic curves. *In ICML*, pages 43–50, 2003.

[BCGJ11] Francesco BONCHI, Carlos CASTILLO, Aristides GIONIS et Alejandro JAIMES : Social Network Analysis and Mining for Business Applications. *ACM Trans. Intell. Syst. Technol.*, 2, 2011.

[BCV12] Yoshua BENGIO, Aaron COURVILLE et Pascal VINCENT : Representation Learning : A Review and New Perspectives. *arXiv*, (1993):1–34, 2012.

[Ber05] Pavel BERKHIN : Survey : A survey on pagerank computing. *Internet Mathematics*, 2(1):73–120, 2005.

[BGLL08] Vincent D BLONDEL, Jean-Loup GUILLAUME, Renaud LAMBIOTTE et Etienne LEFEBVRE : Fast unfolding of communities in large networks. *Journal of Statistical Mechanics : Theory and Experiment*, 2008(10):P10008, 2008.

[BJF11] Felder BÉATRICE et Colin JEAN-FRANÇOIS : Parcours client : la nécessité d'une stratégie multicanal sans rupture. Rapport technique, Orange, 2011.

[BKM11] D J BARTHOLOMEW, M KNOTT et I MOUSTAKI : *Latent Variable Models and Factor Analysis : A Unified Approach*. Wiley Series in Probability and Statistics. Wiley, 2011.

[Bra97] Andrew P BRADLEY : The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recogn.*, 30(7):1145–1159, 1997.

[BST99] Alex BERSON, Stephen SMITH et Kurt THEARLING : *Building Data Mining Applications for CRM*. McGraw-Hill Professional, 1st édition, 1999.

[BSVW04] Tom BRIJS, Gilbert SWINNEN, Koen VANHOOF et Geert WETS : Building an association rules framework to improve product assortment decisions. *Data Mining and Knowledge Discovery*, 8(1):7–23, 2004.

[BV12] Dries F. BENOIT et Dirk VAN DEN POEL : Improving customer retention in financial services using kinship network information. *Expert Systems with Applications*, 39(13):11435–11442, 2012.

[CBP13] J D CRUZ, C BOTHOREL et F POULET : Community Detection and Visualization in Social Networks : Integrating Structural and Semantic Information. *Acm Transactions on Intelligent Systems and Technology*, 5(1), 2013.

[CHH07] Horng-Jinh CHANG, Lun-Ping HUNG et Chia-Ling HO : An anticipation model of potential customers' purchasing behavior based on clustering analysis and association rules analysis, 2007.

[Chu99] F R K CHUNG : Spectral Graph Theory. *ACM SIGACT News*, 30:14, 1999.

[CLEZG12] D. COMBE, C. LARGERON, E. EGYED-ZSIGMOND et M. GERY : Combining relations and text in scientific network clustering. *In Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on*, pages 1248–1253, Aug 2012.

[CM06] Harold CASSAB et Douglas L. MACLACHLAN : Interaction fluency : a customer performance measure of multichannel service, 2006.

[CM11] Irena Pletikosa CVIJIKJ et Florian MICHAHELLES : Monitoring Trends on Facebook. *In Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on*, pages 895–902, 2011.

[CNP03] Gianfranco CHICCO, Roberto NAPOLI et Federico PIGLIONE : Application of clustering algorithms and Self Organising Maps to classify electricity customers. *In 2003 IEEE Bologna PowerTech - Conference Proceedings*, volume 1, pages 373–379, 2003.

[CP01] Gert CAUWENBERGHS et Tomaso POGGIO : Incremental and Decremental Support Vector Machine Learning. *Learning*, 13(13):409, 2001.

[CS09] Kang CAO et Pei Ji SHAO : Customer churn prediction based on SVM-RFE. *In 2008 International Seminar on Business and Information Management, ISBIM 2008*, volume 1, pages 306–309, 2009.

[Cun15] Delphine CUNY : Orange tourne la page de "conquêtes 2015" et dévoile le plan "essentiels 2020". `http://www.latribune.fr/technos-medias/orange-tourne-la-page-de-conquetes-2015-et-devoile-le-plan-essentiels-2020-461324.html`, 2015. Accessed : 2015-03-16.

[CWLL03] Ding An CHIANG, Yi Fan WANG, Shao Lun LEE et Cheng Jung LIN : Goal-oriented sequential pattern for network banking churn analysis. *Expert Systems with Applications*, 25(3):293–302, 2003.

[CYS12] Jiangfeng CHEN, Jianjun YU et Yi SHEN : Towards topic trend prediction on a topic evolution model with social connection. *In Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '12, pages 153–157, Washington, DC, USA, 2012. IEEE Computer Society.

[DBG+09] Gideon DROR, Marc BOULLÉ, Isabelle GUYON, Vincent LEMAIRE et David VOGEL, éditeurs. *Proceedings of KDD-Cup 2009 competition, Paris, France, June 28, 2009*, volume 7 de *JMLR Proceedings*. JMLR.org, 2009.

[DC03] C.P. DIEHL et G. CAUWENBERGHS : SVM incremental learning, adaptation and optimization. *Proceedings of the International Joint Conference on Neural Networks, 2003.*, 4, 2003.

[DDF+90] Scott DEERWESTER, Susan T. DUMAIS, George W. FURNAS, Thomas K. LANDAUER et Richard HARSHMAN : Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[DH01] Pedro DOMINGOS et Geoff HULTEN : Catching up with the data : Research issues in mining data streams. *In In Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2001.

[DK09] Christian DESROSIERS et George KARYPIS : Within-network classification using local structure similarity. *In Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2009, Bled, Slovenia, September 7-11, 2009, Proceedings, Part I*, pages 260–275. Springer, 2009.

[DLP03] Kushal DAVE, Steve LAWRENCE et D.M. PENNOCK : Mining the peanut gallery : Opinion extraction and semantic classification of product reviews. *In Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM, 2003.

[Dom03] Pedro DOMINGOS : Prospects and challenges for multi-relational data mining. *ACM SIGKDD explorations newsletter*, 2003(1):80–83, 2003.

[DR02] Pedro DOMINGOS et Matt RICHARDSON : Mining the Network Value of Customers. *In In Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining*, pages 57–66. ACM Press, 2002.

[DSV+08] K DASGUPTA, R SINGH, B VISWANATHAN, D CHAKRABORTY, S MUKHERJEA, A A NANAVATI et A JOSHI : Social ties and their relevance to churn in mobile telecom networks. *Proceedings of the 11th international conference on Extending database technology Advances in database technology*, pages 668–677, 2008.

[DŽ03] S DŽEROSKI : Multi-relational data mining : an introduction. *ACM SIGKDD Explorations Newsletter*, 5(1):1–16, 2003.

[EK10] David EASLEY et Jon KLEINBERG : *Networks, Crowds, and Markets : Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[FBC+10] Raphaël FÉRAUD, Marc BOULLÉ, Fabrice CLÉROT, Françoise FESSANT et Vincent LEMAIRE : The orange customer analysis platform. *In Proceedings of the 10th Industrial Conference on Advances in Data Mining : Applications and Theoretical Aspects*, ICDM'10, pages 584–594, Berlin, Heidelberg, 2010. Springer-Verlag.

[FCH+08] Rong-En FAN, Kai-Wei CHANG, Cho-Jui HSIEH, Xiang-Rui WANG et Chih-Jen LIN : Liblinear : A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, juin 2008.

[Fre79] L FREEMAN : Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215–239, 1979.

[GDG12] Sheng GAO, Ludovic DENOYER et Patrick GALLINARI : Prédiction de liens temporels en intégrant les informations de contenu et de structure. *Ingénierie des Systèmes d'Information*, 17(6):75–90, 2012.

[GHB09] A GO, L HUANG et R BHAYANI : Sentiment Analysis of Twitter Data. *Entropy*, 2009(June):17, 2009.

[GKKL13] Nadav Golbandi GOLBANDI, Liran Katzir KATZIR, Yehuda Koren KOREN et Ronny Lempel LEMPEL : Expediting search trend detection via

prediction of query counts. *In Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 295–304, New York, NY, USA, 2013. ACM.

[GL09] Vishal GUPTA et Gurpreet S LEHAL : A Survey of Text Mining Techniques and Applications. *Journal of Emerging Technologies in Web Intelligence*, 1(1):60–76, 2009.

[Gre09] Paul GREENBERG : Social CRM Comes of Age. Rapport technique, Oracle, 2009.

[GT07] L GETOOR et B TASKAR : *Introduction to Statistical Relational Learning*, volume L de *Adaptive computation and machine learning*. MIT Press, 2007.

[GTM11] Neil Zhenqiang GONG, Ameet TALWALKAR et Lester MACKEY : Jointly Predicting Links and Inferring Attributes using a Social-Attribute Network (SAN). *CoRR*, abs/1112.3, décembre 2011.

[GXH+12] Neil Zhenqiang GONG, Wenchang XU, Ling HUANG, Prateek MITTAL, Emil STEFANOV, Vyas SEKAR et Dawn SONG : Evolution of social-attribute networks : Measurements, modeling, and implications using google+. *In Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, IMC '12, pages 131–144, New York, NY, USA, 2012. ACM.

[HB07] Carine HUE et Marc BOULLÉ : A new probabilistic approach in rank regression with optimal bayesian partitioning. *J. Mach. Learn. Res.*, 8: 2727–2754, décembre 2007.

[HCSZ06] Mohammad Al HASAN, Vineet CHAOJI, Saeed SALEM et Mohammed ZAKI : Link prediction using supervised learning. *SDM'06 : Workshop on Link . . .*, 2006.

[Hec08] David HECKERMAN : A tutorial on learning with Bayesian networks. *Innovations in Bayesian Networks*, pages 301–354, 2008.

[HK12] Taeho HONG et Eunmi KIM : Segmenting customers in online stores based on factors that affect the customer's intention to purchase. *In Expert Systems with Applications*, volume 39, pages 2127–2131, 2012.

[HL04] Minqing HU et Bing LIU : Mining Opinion Features in Customer Reviews. *Science*, 21(2):755–760, 2004.

[HRH02] P. D HOFF, A. E RAFTERY et M. S HANDCOCK : Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.

[Hua06] Zan HUANG : Link Prediction Based on Graph Topology : The Predictive Value of Generalized Clustering Coefficient. *In Proceedings of the Workshop on Link Analysis : Dynamics and Static of Large Networks (LinkKDD2006)*, pages 1–31. ACM, 2006.

[IBM11] IBM : From social media to Social CRM. *Business*, 2011.

[IRK12] Adnan IDRIS, Muhammad RIZWAN et Asifullah KHAN : Churn prediction in telecom using Random Forest and PSO based data balancing in combination with various feature selection strategies. *Computers and Electrical Engineering*, 38(6):1808–1819, 2012.

[JA07] Pawel JURCZYK et Eugene AGICHTEIN : HITS on Question Answer Portals : Exploration of Link Analysis for Author Ranking. *Evaluation*, pages 845–846, 2007.

[JDG14] Yann JACOB, Ludovic DENOYER et Patrick GALLINARI : Learning latent representations of nodes for classifying in heterogeneous social networks. *In Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 373–382, New York, NY, USA, 2014. ACM.

[JK12] Prachi JOSHI et Parag KULKARNI : Incremental Learning : Areas and Methods–A Survey. *International Journal of Data Mining & Knowledge Management Process*, 2(5), 2012.

[JN02] David JENSEN et Jennifer NEVILLE : Autocorrelation and linkage cause bias in evaluation of relational learners. *In Proceedings of the Twelfth International Conference on Inductive Logic Programming*, pages 101–116. Springer, 2002.

[JN06] Nenad JUKIC et Svetlozar NESTOROV : Comprehensive data warehouse exploration with qualified association-rule mining. *Decision Support Systems*, 42(2):859–878, 2006.

[JNG04] David JENSEN, Jennifer NEVILLE et Brian GALLAGHER : Why collective inference improves relational classification. *In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 593–598, New York, NY, USA, 2004. ACM.

[KAD+10] V KUMAR, L AKSOY, B DONKERS, R VENKATESAN, T WIESEL et S TILLMANNS : Undervalued or Overvalued Customers : Capturing Total Customer Engagement Value. *Journal of Service Research*, 13(3):297–310, 2010.

[KBV09] Yehuda KOREN, Robert BELL et Chris VOLINSKY : Matrix Factorization Techniques for Recommender Systems. *Computer*, 42(8):30–37, août 2009.

[KF07] Daphne KOLLER et Nir FRIEDMAN : Graphical Models in a Nutshell. *In* Lise GETOOR et Ben TASKAR, éditeurs : *Introduction to statistical relational learning*, pages 13–54. MIT Press, 2007.

[KGP+04] April KONTOSTATHIS, LeonM. GALITSKY, WilliamM. POTTENGER, Soma ROY et DanielJ. PHELPS : A survey of emerging trend detection in textual data mining. *In* MichaelW. BERRY, éditeur : *Survey of Text Mining*, pages 185–224. Springer New York, 2004.

[Kle99] Jon M KLEINBERG : Hubs, authorities, and communities. *ACM Comput. Surv.*, 31(4), 1999.

[KN11] Brian KARRER et M. E J NEWMAN : Stochastic blockmodels and community structure in networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 83(1), 2011.

[LC06] Aurélie LEMMENS et Christophe CROUX : Bagging and Boosting Classification Trees to Predict Churn, 2006.

[LCCL06] Tian Shyug LEE, Chih Chou CHIU, Yu Chao CHOU et Chi J. LU : Mining the customer credit using classification and regression tree and multivariate adaptive regression splines. *Computational Statistics and Data Analysis*, 50(4):1113–1130, 2006.

[LGK+10] Yucheng LOW, Joseph GONZALEZ, Aapo KYROLA, Danny BICKSON, Carlos GUESTRIN et Joseph M. HELLERSTEIN : Graphlab : A new framework for parallel machine learning. *CoRR*, abs/1006.4990, 2010.

[Li10] WJ LI : *Latent factor models for statistical relational learning*. Thèse de doctorat, The Hong Kong University of Science and Technology, 2010.

[LIT92] Pat LANGLEY, Wayne IBA et Kevin THOMPSON : An analysis of Bayesian classifiers. *In Proceedings of the National Conference on Artificial Intelligence*, pages 223–223. Citeseer, 1992.

[LK07] D LIBEN NOWELL et Jon KLEINBERG : The link prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, mai 2007.

[LK10] G. LEFAIT et T. KECHADI : Customer segmentation architecture based on clustering techniques. *In Digital Society, 2010. ICDS '10. Fourth International Conference on*, pages 243–248, Feb 2010.

[LP09] Y W LO et V POTDAR : A review of opinion mining and sentiment classification framework in social networks. *2009 3rd IEEE International Conference on Digital Ecosystems and Technologies*, pages 396–401, 2009.

[LTBC14] Duc Kinh LE TRAN, Cécile BOTHOREL et Pascal CHEUNG-MON-CHAN : Incremental learning with latent factor models for attribute prediction in social-attribute networks. *In 14èmes Journées Francophones Extraction et Gestion des Connaissances, EGC 2014, Rennes, France, 28-32 Janvier, 2014*, pages 77–82, 2014.

[LTBCK14] Duc Kinh LE TRAN, Cécile BOTHOREL, Pascal CHEUNG-MON-CHAN et Yvon KERMARREC : Incremental learning with social media data to predict near real-time events. *In Discovery Science - 17th International Conference, DS 2014, Bled, Slovenia, October 8-10, 2014. Proceedings*, pages 180–191, 2014.

[Lux07] Ulrike LUXBURG : A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[LV05] Bart LARIVIERE et Dirk VAN DEN POEL : Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert Systems with Applications*, 29(2):472–484, 2005.

[LW09] H W LAM et Chen WU : Finding Influential eBay Buyers for Viral Marketing A Conceptual Model of BuyerRank. *In Advanced Information Networking and Applications, 2009. AINA '09. International Conference on*, pages 778–785, 2009.

[LY09] WJ LI et DY YEUNG : Relation regularized matrix factorization. *In IJCAI-09*, IJCAI'09, pages 1126–1131, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

[MG10] Maria MERCANTI-GUÉRIN : Analyse des réseaux sociaux et communautés en ligne : quelles applications en marketing. *Management & Avenir*, 2(32):132–153, 2010.

[MK10] Michael MATHIOUDAKIS et Nick KOUDAS : TwitterMonitor : trend detection over the twitter stream. *Proceedings of the 2010 international conference on Management of data*, pages 1155–1158, 2010.

[MP07] SA MACSKASSY et Foster PROVOST : Classification in networked data : A toolkit and a univariate case study. *The Journal of Machine Learning Research*, 8(December 2004):935–983, 2007.

[MQ09] Hongxia MA et Min QIN : Research method of customer churn crisis based on decision tree. *In Proceedings - International Conference on Management and Service Science, MASS 2009*, 2009.

[MRMCMVUnL14] Arturo MONTEJO-RÁEZ, Eugenio MARTÍNEZ-CÁMARA, M. Teresa MARTÍN-VALDIVIA et L. Alfonso Ureña LÓPEZ : Ranked wordnet graph for sentiment polarity classification in twitter. *Comput. Speech Lang.*, 28(1):93–107, janvier 2014.

[New06] M E J NEWMAN : Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 103(23):8577–8582, 2006.

[NS01] Krzysztof NOWICKI et Tom A. B SNIJDERS : Estimation and Prediction for Stochastic Blockstructures, 2001.

[NXC09] E.W.T. NGAI, Li XIU et D.C.K. CHAU : Application of data mining techniques in customer relationship management : A literature review and classification. *Expert Systems with Applications*, 36(2):2592–2602, mars 2009.

[PBMW99] Lawrence PAGE, Sergey BRIN, Rajeev MOTWANI et Terry WINOGRAD : The PageRank Citation Ranking : Bringing Order to the Web, 1999.

[PE05] Ana-Maria POPESCU et Oren ETZIONI : Extracting product features and opinions from reviews. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing HLT 05*, 5(October):339–346, 2005.

[PL08] Bo PANG et Lillian LEE : Opinion Mining and Sentiment Analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, 2008.

[PP10] Alexander PAK et Patrick PAROUBEK : Twitter as a corpus for sentiment analysis and opinion mining. *In Proceedings of LREC*, volume 2010, pages 1320–1326, 2010.

[PS02] Atul PARVATIYAR et Jagdish N. SHETH : Customer relationship management : emerging practice, process and discipline. *Journal of Economic and Social Research*, 3:6–23, 2002.

[PU03] Alexandrin POPESCUL et Lyle H UNGAR : Statistical Relational Learning for Link Prediction. *Information Sciences*, 149:172, 2003.

[Raj11] Sankar RAJAGOPAL : Customer data clustering using data mining technique. *CoRR*, abs/1112.2663, 2011.

[RFP12] Yiye RUAN, David FUHRY et Srinivasan PARTHASARATHY : Efficient Community Detection in Large Networks using Content and Links. *arXiv preprint arXiv :1212.0146*, pages 1–21, 2012.

[RK03] Luc De RAEDT et Kristian KERSTING : Probabilistic Logic Learning. *SIGKDD Explor. Newsl.*, 5(1):31–48, 2003.

[RWY02] Chris RYGIELSKI, Jyun-Cheng WANG et David C. YEN : Data mining techniques for customer relationship management. *Technology in Society*, 24(4):483 – 502, 2002.

[Sal12] Christophe SALPERWYCK : *Apprentissage incrémental en ligne sur flux de données*. Thèse de doctorat, 2012. Thèse de doctorat dirigée par Preux, Philippe Informatique Lille 3 2012.

[SG08] AP SINGH et GJ GORDON : Relational learning via collective matrix factorization. *In Proceeding of the 14th ACM SIGKDD*, numéro June, pages 650–658, New York, NY, USA, 2008. ACM.

[Shl05] Jonathon SHLENS : A Tutorial on Principal Component Analysis. *Measurement*, 51:52, 2005.

[SL10] Christophe SALPERWYCK et Vincent LEMAIRE : Classification incrémentale supervisée : un panel introductif. *In AAFD*, pages 121–148, 2010.

[SN13] Marie Laure SOUBILS et Ludovic NODIER : Observatoire des Services Clients 2013. Rapport technique, BVA Group, Viséo Conseil, 2013.

[TA07] Ben TASKAR et Pieter ABBEEL : Relational Markov Networks. *In* Lise GETOOR et Ben TASKAR, éditeurs : *Introduction to Statistical Relational Learing*, pages 175–199. MIT Press, 2007.

[TAK02] Ben TASKAR, Pieter ABBEEL et Daphne KOLLER : Discriminative probabilistic models for relational data. *In UAI'02 Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, UAI'02, pages 485–492, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[TL11] Lei TANG et Huan LIU : Leveraging social media networks for classification. *Data Mining and Knowledge Discovery*, 23(3):447–478, janvier 2011.

[TSK01] Ben TASKAR, Eran SEGAL et Daphne KOLLER : Probabilistic Classification and Clustering in Relational Data. *In Proceedings of the 17th international joint conference on Artificial intelligence - Volume 2*, pages 870–876, Seattle, WA, USA, 2001. Morgan Kaufmann Publishers Inc.

[TWAK03] Ben TASKAR, Ming-fai WONG, Pieter ABBEEL et Daphne KOLLER : Link Prediction in Relational Data. *In Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*. MIT Press, 2003.

[UU89] Paul E. UTGOFF et Paul UTGOFF : Incremental Induction of Decision Trees. *Machine Learning*, 4(2):161–186, 1989.

[VB05] Dirk VAN DEN POEL et Wouter BUCKINX : Predicting online-purchasing behaviour. *European Journal of Operational Research*, 166(2): 557–575, 2005.

[VCL11] Régine VANHEEMS et Isabelle COLLIN-LACHAUD : Comment le parcours cross-canal du consommateur transformet-il son expérience de shopping. *Actes du 14ème Colloque Etienne Thil, Lille-Roubaix*, 11-13, 2011.

[VD01] Peter C. VERHOEF et Bas DONKERS : Predicting customer potential value an application in the insurance industry. *Decision Support Systems*, 32(2):189–199, 2001.

[WC02] Chih Ping WEI et I. Tang CHIU : Turning telecommunications call details to churn prediction : A data mining approach. *Expert Systems with Applications*, 23(2):103–112, 2002.

[WF94] Stanley WASSERMAN et Katherine FAUST : *Social Network Analysis : Methods and Applications*. Numéro 8 de Structural analysis in the social sciences. Cambridge University Press, 1 édition, 1994.

[WLJH10] Jianshu WENG, Ee-Peng LIM, Jing JIANG et Qi HE : TwitterRank : finding topic-sensitive influential twitterers. *In Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 261–270, New York, NY, USA, 2010. ACM.

[Wor08] Jennifer WORTMAN : Viral Marketing and the Diffusion of Trends on Social Networks. *Science*, 2008.

[WSLC12] Shruti WAKADE, Chandra SHEKAR, Kathy J LISZKA et Chien-chung CHAN : Text Mining for Sentiment Analysis of Twitter Data. *IKE*, 2012.

[XJ08] Guo-en XIA et Wei-dong JIN : Model of Customer Churn Prediction on Support Vector Machine, 2008.

[YGWH10] Zhijun YIN, Manish GUPTA, Tim WENINGER et Jiawei HAN : A Unified Framework for Link Recommendation Using Random Walks. ASONAM '10, pages 152–159, Washington, DC, USA, 2010. IEEE Computer Society.

[ZGGK09] Elena ZHELEVA, Lise GETOOR, Jennifer GOLBECK et Ugur KUTER : Using friendship ties and family circles for link prediction. *In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 5498 LNAI, pages 97–113, 2009.

[ZGH09] Xiao Bin ZHANG, Feng GAO et Hui HUANG : Customer-churn research based on customer segmentation. *In Proceedings - 2009 International Conference on Electronic Commerce and Business Intelligence, ECBI 2009*, pages 443–446, 2009.

[ZWSP08] Yunhong ZHOU, Dennis WILKINSON, Robert SCHREIBER et Rong PAN : Large-Scale Parallel Collaborative Filtering for the Netflix Prize. AAIM '08, pages 337–348, Berlin, Heidelberg, 2008. Springer-Verlag.

[ZYCG07] Shenghuo ZHU, Kai YU, Yun CHI et Yihong GONG : Combining content and link for classification using matrix factorization. *In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 487–494. ACM, 2007.

Résumé

Cette thèse d'informatique en fouille de données et apprentissage automatique s'inscrit dans le contexte applicatif de la gestion de la relation client (Customer Relationship Management ou CRM). Avec l'émergence des média sociaux, les entreprises perçoivent actuellement la nécessité d'une stratégie de relation client intercanale dans laquelle elles suivent le parcours du client sur l'ensemble des canaux d'interactions tels que les média sociaux, la hot line... et cela de manière integrée. L'objectif applicatif de la thèse est de concevoir de nouvelles techniques permettant de prédire les comportements du client à partir des données issues de ces multiples canaux. Nous nous intéressons aux comportements qui caractérisent l'engagement du client vis-à-vis de l'entreprise. Nous effectuons d'abord une analyse des besoins dans laquelle nous montrons la nécessité des nouvelles techniques de fouilles de données pour une stratégie de relation client intégrant plusieurs canaux de nature différente. Nous introduisons ensuite une nouvelle méthode d'apprentissage incrémental basée sur les modèles à facteurs latents et sur la représentation de réseau social attribué. Nous effectuons ensuite des expérimentations sur des données synthétiques et réelles. Nous montrons que notre méthode de réduction de dimension est capable d'extraire des variables latentes informatives pour prédire les comportements des clients à partir de données intercanales. Dans les perspectives, nous proposons quelques pistes d'amélioration de notre méthode, notamment d'autres modèles à facteurs latents permettant d'exploiter différents types de corrélations entre les individus dans le graphe social.

Mots-clés : Customer Relationship Management, Stratégie cross canal, Fouille de media sociaux, Modèle à facteurs latents, Factorisation de matrice, Analyse de réseaux sociaux, Graph mining

Abstract

This thesis is in the field of data mining and in the context of Customer Relationship Management (CRM). With the emergence of social media, companies today have seen the need for an interchannel (or cross-channel) strategy in which they keep track of their clients' histories through a consistent combination of multiple channels. The goal of this thesis is to develop new data mining methods which allow predicting customer behaviors using data collected from multiple channels such as social media, call center... We are interested in all types of customer behaviors that characterized their engagement with respect to the company. First of all, we perform a needs analysis in terms of data mining for interchannel CRM strategy. Next, we propose a new method of prediction of customer behaviors in the context of interchannel CRM. In our method, we use a social attributed network to represent the data from multiple channels and perform incremental learning based on latent factor models. We then carry out experiments on both synthetic and real data. We show that our method based on the latent factor models is capable of leveraging informative latent factors from interchannel data. In future works, we consider some ways to improve the performance of our method, especially latent factor models that are able to leverage different types of relational correlation between individuals in the social graph.

Keywords : Customer Relationship Management, Cross-channel CRM, Social media mining, Latent factor models, Matrix factorization, Social network analysis, Graph mining

TELECOM Bretagne

n° d'ordre : 2015telb0353

Télécom Bretagne

Technopôle Brest-Iroise - CS 83818 - 29238 Brest Cedex 3

Tél : + 33(0) 29 00 11 11 - Fax : + 33(0) 29 00 10 00